

OPTIONS NODECK,LIST,XREF,NOREL,OBJ(P)

THE LIST OF OPTIONS USED DURING THIS ASSEMBLY IS-- NODECK,LIST,XREF,NOREL,OBJ



ERR LOC OBJECT CODE				ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20 PAGE 2			
0000					1	#INSTD	START 0				
					2		PRINT ON,NODATA				
					3	*	@SYS EXP-N				
					214+		PRINT ON				
					215	*	@HDW EXP-N				
					400+		PRINT ON				
					401	*	@FXD EXP-N				
					806+		PRINT ON				
					807	*	@ERM EXP-N				
					1429+		PRINT ON				
					1430	*	@B@E EXP-N				
					2330+		PRINT ON				
					2331	*	\$I@E EXP-N,PREC-S				
					2492+		PRINT ON				
					2493	*	\$I\$E EXP-N				
					2647+		PRINT ON				
					2648	*	\$V\$E EXP-N				
					3071+		PRINT ON				

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20	PAGE	3
		3073		*****				
		3074	*	5703-XM1 COPYRIGHT IBM CORP. 1970				*
		3075	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083				*
		3076	*					*
		3077		*****				
		3078	*	STATUS				*
		3079	*	VERSION 1 MODIFICATION 0				*
		3080	*					*
		3081	*	FUNCTION				*
		3082	*	* IMINIT MODIFIES THE CORE-RESIDENT BASIC INTERPRETER FOR AN				*
		3083	*	EXPANDED CORE CONFIGURATION, INITIALIZES THE CORE VIRTUAL				*
		3084	*	MEMORY PAGE REGIONS, AND SETS RUN-TIME INDICATORS PRIOR TO				*
		3085	*	ENTERING THE INTERPRETER EXECUTIVE ROUTINE.				*
		3086	*	* CORE PAGE REGION EXPANSION - PAGING ROUTINE INSTRUCTIONS AND				*
		3087	*	TABLE SIZES ARE MODIFIED TO REFLECT THE NUMBER OF CORE PAGES				*
		3088	*	AVAILABLE IN EXCESS OF 8K. EXPANDED TABLES ARE INITIALIZED				*
		3089	*	TO BINARY ZEROS.				*
		3090	*	* CORE PAGE REGION INITIALIZATION - DISK VIRTUAL MEMORY PAGES,				*
		3091	*	BEGINNING WITH PAGE 0, ARE LOADED INTO THE CORE PAGE REGION.				*
		3092	*	THE PAGE REFERENCE TABLE IN IPGMDL IS INITIALIZED TO REFLECT				*
		3093	*	THIS STARTING CORE PAGE ARRANGEMENT.				*
		3094	*	* FLOW TRACE CAPABILITY - THE INTERPRETER STATEMENT HEADER PMC				*
		3095	*	EXECUTION ROUTINE IN INTERP IS SET TO PERMIT TRACE FLOW WHEN				*
		3096	*	SYSTEM INDICATOR \$TRACE HAS BEEN SET ON.				*
		3097	*	* PROGRAM 'DATA' FILE POINTERS - THE 'DATA' FILE POINTERS ARE SET				*
		3098	*	TO REFERENCE THE FIRST ELEMENT IN THE PROGRAM 'DATA' FILE.				*
		3099	*	* MASKED INQUIRY REQUEST - CONSOLE INTERRUPT IS MASKED FOR RUN-				*
		3100	*	TIME UTILIZATION ONLY AT SPECIFIC POINTS DURING EXECUTION.				*
		3101	*	* SYSTEM INDICATORS -				*
		3102	*	* \$VMDEF (V.M. DEFINITION) - SET ON IN \$XIND1.				*
		3103	*	* \$EXCMD (EXECUTION MODE) - SET ON IN \$XIND2.				*
		3104	*					*
		3105	*	ENTRY POINTS				*
		3106	*	IMINIT IS THE LABEL ASSOCIATED WITH THE SINGLE INTERPRETER				*
		3107	*	INITIATOR ENTRY POINT, WHICH IS IDENTICALLY THE INTERPRETER				*
		3108	*	ENTRY POINT (#INSTD OR #INLNG). CALLING SEQUENCE FOR INTERPRETER				*
		3109	*	LOADING AND EXECUTION IS:				*
		3110	*	B \$RLOAD				*
		3111	*	DC AL2(DPLADR)				*
		3112	*	WHERE DPLADR IS THE LABEL ASSOCIATED WITH THE #INSTD OR #INLNG				*
		3113	*	LOADING DISK PARAMETER LIST. IMINIT ENTRY IS SUBJECT TO THE				*
		3114	*	INPUT CONDITIONS DESCRIBED BELOW.				*
		3115	*					*
		3116	*	INPUT				*
		3117	*	* \$EXFTR - 1 BYTE, FOR THE SYSTEM CORE EXTENSION FACTOR. THIS				*
		3118	*	CONTAINS THE NUMBER OF CORE PAGES, IN EXCESS OF 8K, AVAILABLE				*
		3119	*	FOR PAGING OPERATIONS.				*
		3120	*	* \$INLNO - 2 BYTES, FOR THE EXECUTION LINE NUMBER. THIS IS USED				*
		3121	*	COMMUNICATION PARAMETER FROM THE COMPILER, AND CONTAINS				*
		3122	*	THE VIRTUAL ADDRESS OF THE FIRST 'DCA' PSEUDO INSTRUCTION				*
		3123	*	GENERATED IN VIRTUAL MEMORY FOR THE PROGRAM 'DATA' FILE.				*
		3124	*	* \$XIND1 - 1 BYTE, FOR SYSTEM EXECUTION INDICATOR 1. IMINIT				*
		3125	*	TESTS A SINGLE BIT (\$TFLOW) WITHIN THIS BYTE. WHEN THIS BIT				*
		3126	*	IS ON, FLOW TRACE MODE IS INDICATED.				*
		3127	*					*
		3128	*	OUTPUT				*

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 4
		3129	*	* CORE PAGE REGION - THIS CONTAINS (N) 256-BYTE PAGES, WHERE	*
		3130	*	* N = S (FOR \$EXFTR EQUAL 0)	*
		3131	*	* N = S+X-1 (FOR \$EXFTR GREATER THAN 0)	*
		3132	*	(S) IS THE NUMBER OF CORE PAGES ALLOCATED FOR AN 8K SYSTEM, AND	*
		3133	*	(X) IS THE CURRENT VALUE IN \$EXFTR. THE REGION IS LOADED WITH	*
		3134	*	VIRTUAL PAGES (0) THROUGH (N-1) BEGINNING AT THE LOWEST CORE	*
		3135	*	LOCATION.	*
		3136	*	* PAGE INDICATOR TABLE - THIS IPGMDL TABLE NORMALLY CONTAINS (S)	*
		3137	*	1-BYTE ENTRIES, AND IS EXPANDED TO CONTAIN (S+X-1) ENTRIES WHEN	*
		3138	*	\$EXFTR IS GREATER THAN 0. EXPANDED ENTRY LOCATIONS ARE SET	*
		3139	*	TO BINARY ZEROS.	*
		3140	*	* PAGE USAGE VALUE TABLE - THIS IPGMDL TABLE NORMALLY CONTAINS	*
		3141	*	(S) 2-BYTE ENTRIES, AND IS EXPANDED TO CONTAIN (S+X-1) ENTRY	*
		3142	*	LOCATIONS WHEN \$EXFTR IS GREATER THAN 0. EXPANDED ENTRY LOCA-	*
		3143	*	TIONS ARE SET TO BINARY ZEROS.	*
		3144	*	* IPGMDL CODING - PAGING MODULE MACHINE INSTRUCTIONS ARE MODIFIED	*
		3145	*	TO REFLECT CORE PAGE EXPANSION WHEN \$EXFTR IS GREATER THAN ZERO.	*
		3146	*	* PAGE REFERENCE TABLE - THIS IPGMDL TABLE CONTAINS 256 1-BYTE	*
		3147	*	ENTRIES, EACH OF WHICH REFERENCES A PARTICULAR PAGE IN VIRTUAL	*
		3148	*	MEMORY. THE FIRST (N) TABLE ENTRIES ARE INITIALIZED WITH PRO-	*
		3149	*	PRIATE CORE PAGE NUMBERS TO REFLECT CORE PAGE REGION LOADING.	*
		3150	*	* IZTFSW - 1 BYTE, FOR THE INTERPRETER TRACE FLOW SWITCH. THIS	*
		3151	*	SWITCH IS SET TO CODE @NOP WHEN 'TRACE FLOW' HAS BEEN SPECIFIED	*
		3152	*	(\$TFLOW IN \$XIND1 IS ON).	*
		3153	*	* IZDATA - 2 BYTES, FOR THE PROGRAM INTERNAL 'DATA' FILE POINTER.	*
		3154	*	THIS IS SET TO THE VALUE LOADED INTO \$INLNO AT COMPILE-TIME.	*
		3155	*	* IZDAT1 - 2 BYTES, FOR THE PROGRAM INTERNAL 'DATA' FILE BASE	*
		3156	*	POINTER. THIS IS ALSO SET TO THE VALUE LOADED INTO \$INLNO AT	*
		3157	*	COMPILE-TIME.	*
		3158	*	* \$INLNO - 2 BYTES, FOR THE EXECUTION LINE NUMBER. THIS IS SET	*
		3159	*	TO AN INVALID LINE NUMBER (X'FFFF') TO INDICATE THE START OF	*
		3160	*	PROGRAM EXECUTION.	*
		3161	*	* \$XIND1 - 1 BYTE, FOR SYSTEM EXECUTION INDICATOR 1. BIT \$VMDEF	*
		3162	*	IN THIS BYTE IS SET TO B'1' TO INDICATE THAT VIRTUAL MEMORY HAS	*
		3163	*	BEEN DEFINED FOR EXECUTION.	*
		3164	*	* \$XIND2 - 1 BYTE, FOR SYSTEM EXECUTION INDICATOR 2. BIT \$EXCMD	*
		3165	*	IN THIS BYTE IS SET TO B'1' TO INDICATE EXECUTION MODE.	*
		3166	*	* \$CIMSK - 1 BYTE, FOR THE CONSOLE INTERRUPT MASK FIELD. THIS IS	*
		3167	*	SET TO CODE @NOP TO MASK (DISABLE) INQUIRY REQUEST FROM THE	*
		3168	*	SYSTEM CONSOLE.	*
		3169	*		*
		3170	*	*EXTERNAL REFERENCES	*
		3171	*	* \$DISKN - ENTRY POINT FOR THE SYSTEM PHYSICAL DISK IOCS.	*
		3172	*	* \$WAITF - CORE ADDRESS OR 'WAIT' FUNCTION DISK PARAMETER LIST.	*
		3173	*	* \$INLNO - 2 BYTES, FOR THE EXECUTION LINE NUMBER.	*
		3174	*	* \$FXFTR - 1 BYTE, FOR THE SYSTEM CORE EXTENSION FACTOR.	*
		3175	*	* \$XIND1 - 1 BYTE, FOR SYSTEM EXECUTION INDICATOR 1.	*
		3176	*	* \$XIND2 - 1 BYTE, FOR SYSTEM EXECUTION INDICATOR 2.	*
		3177	*	* \$CIMSK - 1 BYTE, FOR THE INQUIRY REQUEST MASK BYTE.	*
		3178	*	* I\$CPG1 - CORE ADDRESS FOR 8K SYSTEM CORE PAGE REGION 1ST BYTE.	*
		3179	*	* IZDATA - 2 BYTES, FOR THE PROGRAM 'DATA' FILE POINTER.	*
		3180	*	* IZDAT1 - 2 BYTES, FOR THE PROGRAM 'DATA' FILE 1ST ELEMENT PT.	*
		3181	*	* IZTFSW - 1 BYTE, FOR THE INTERPRETER FLOW TRACE SWITCH.	*
		3182	*	* IZPGTB - BASE CORE ADDRESS FOR IPGMDL PAGE REFERENCE TABLE.	*
		3183	*	* INTERP - ENTRY POINT FOR INTERPRETER EXECUTIVE MODULE.	*
		3184	*	* IPGSZ1 - PAGING MODULE REFERENCE TO MAXIMUM CORE SIZE.	*

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 5
		3185	*	* IPGSZ2 - PAGING MODULE REFERENCE TO MAXIMUM CORE SIZE.	*
		3186	*	* IPGSZ3 - PAGING MODULE REFERENCE TO MAXIMUM CORE SIZE.	*
		3187	*	* IPGUT1 - PAGING MODULE REFERENCE TO PAGE USAGE VALLE TABLE.	*
		3188	*	* IPGUT2 - PAGING MODULE REFERENCE TO PAGE USAGE VALUE TABLE.	*
		3189	*	* IPGMX1 - PAGING MODULE REFERENCE TO MAXIMUM NO. OF CORE PAGES.	*
		3190	*	* IPGMX2 - PAGING MODULE REFERENCE TO MAXIMUM NO. OF CORE PAGES.	*
		3191	*	* IPGMX3 - PAGING MODULE REFERENCE TO MAXIMUM NO. OF CORE PAGES.	*
		3192	*	* IPGMX4 - RAGING MODULE REFERENCE TO MAXIMUM NO. OF CORE PAGES.	*
		3193	*	* IPGMX5 - PAGING MODULE REFERENCE TO MAXIMUM NO. OF CORE PAGES.	*
		3194	*		*
		3195	*	*EXITS, NORMAL	*
		3196	*	CONTROL IS PASSED TO INTERPRETER EXECUTIVE ROUTINE INTERP AFTER	*
		3197	*	INITIALIZATION HAS BEEN PERFORMED.	*
		3198	*		*
		3199	*	*EXITS, ERROR	*
		3200	*	N/A	*
		3201	*		*
		3202	*	*TABLES/WORK AREAS	*
		3203	*	* IMINIT CONTAINS NO SIGNIFICANT WORK AREA AS SUCH. HOWEVER,	*
		3204	*	THE MODULE DOES INCLUDE CODING WHICH ESTABLISHES THE FUNCTION	*
		3205	*	EXECUTION WORK AREA AND THE RUN-TIME STACK, BOTH OF WHICH OVER-	*
		3206	*	LAY IMINIT AFTER INITIALIZATION HAS BEEN COMPLETED.	*
		3207	*	* FUNCTION WORK AREA - 50 BYTES, FOR RUN-TIME OPERATIONS INVOLV-	*
		3208	*	ING ARITHMETIC FUNCTIONS. THIS WORK AREA BEGINS IMMEDIATELY	*
		3209	*	FOLLOWING THE 7-BYTE INTERPRETER PROGRAM HEADER.	*
		3210	*	* RUN-TIME STACK - 240 BYTES, FOR GENERAL PMC EXECUTION OPERA-	*
		3211	*	TIONS. THE STACK REGION BEGINS JUST AFTER THE FUNCTION WORK	*
		3212	*	AREA, SO THAT 41 OF THE 240 BYTES FALL IN THE NEXT 'PAGE'.	*
		3213	*	THIS ARRANGEMENT PERMITS THE RUN-TIME STACK POINTER (IZSTAK)	*
		3214	*	TO REQUIRE ADJUSTMENT IN THE DISPLACEMENT BYTE ONLY, SO THAT	*
		3215	*	ELEMENT STACKING BEYOND X'06FF' CAUSES 'STACK OVERFLOW'. THE	*
		3216	*	REGION BETWEEN X'0700' AND X'0728' ACTS AS A BUFFER FOR ARITH-	*
		3217	*	METIC OPERATIONS PERFORMED WHEN THE STACK IS FILLED TO X'06FF'	*
		3218	*	WITH ACTIVE DATA ELEMENTS.	*
		3219	*		*
		3220	*	*ATTRIBUTES	*
		3221	*	RELOCATABLE	*
		3222	*		*
		3223	*	*CHARACTER CODE DEPENDENCY	*
		3224	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*
		3225	*	REPRESENTATION OF THE EVTERNAL CHARACTER SET.	*
		3226	*		*
		3227	*	*NOTES	*
		3228	*	ERROR PROCEDURES	*
		3229	*	N/A	*
		3230	*		*
		3231	*	REGISTER USAGE	*
		3232	*	* REGISTER @BR IS NOT SAVED. IT IS USED FOR IMINIT BASE	*
		3233	*	ADDRESSABILITY, AND RETAINS THE IMINIT BASE ADDRESS AT EXIT.	*
		3234	*	* REGISTER @XR IS NOT SAVED. IT IS USED AS A GENERAL PURPOSE	*
		3235	*	INDEX FOR THE VARIOUS IMINIT OPERATIONS.	*
		3236	*		*
		3237	*	SAVED/RESTORED AREAS	*
		3238	*	N/A	*
		3239	*		*
		3240	*	MODIFICATION CONSIDERATIONS	*

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 6
		3241	*	IMINIT PERFORMS SPECIFIC CORE-EXPANSION DIRECTED MODIFICATIONS	*
		3242	*	ON THE INTERPRETER PAGING CONTROL MODULE (IPGMDL). CODING	*
		3243	*	CHANGES WITHIN IPGMDL MUST BE CONDUCTED SUCH THAT A CONSISTENT	*
		3244	*	RELATIONSHIP IS MAINTAINED.	*
		3245	*		*
		3246	*	REQUIRED MODULES	*
		3247	*	* @SYSEQ - COMMON SYSTEM EQUATES.	*
		3248	*	* @FXDEQ - SYSTEM NUCLEUS ADDRESSES AND INDICATOR EQUATES.	*
		3249	*	* \$B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.	*
		3250	*	* \$I\$EQU - INTERPRETER FIXED LOCATION ADDRESS EQUATES.	*
		3251	*	* \$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD. PREC. ONLY).	*
		3252	*	* \$I@LEQ - INTERPRETER PARAMETER EQUATES (FOR LONG PREC. ONLY).	*
		3253	*	* INTERP - INTERPRETER EXECUTIVE ROUTINE.	*
		3254	*	* IPGMDL - INTERPRETER PAGING CONTROL MODULE.	*
		3255	*	* IZCOMN - INTERPRETER COMMON ADDRESS REFERENCE EQUATES.	*
		3256	*		*
		3257	*	OTHER	*
		3258	*	MODULE IMINIT CONTAINS CODING WHICH DEFINES THE CORE-RESIDENT	*
		3259	*	INTERPRETER PATCH AREA. THIS PATCH AREA IMMEDIATELY FOLLOWS	*
		3260	*	THE RUN-TIME STACK, AND IS USED SUCH THAT THE PAGING MODULE	*
		3261	*	TABLES END EXACTLY AT THE BYTE PRECEDING THE 8K SYSTEM CORE	*
		3262	*	PAGE REGION. ANY CHANGES IN CORE-RESIDENT INTERPRETER MODULES	*
		3263	*	(FOLLOWING IMINIT) WHICH CHANGE THE SIZE OF THE CORE-RESIDENT	*
		3264	*	INTERPRETER REQUIRES THAT THIS PATCH AREA BE MODIFIED SO THAT	*
		3265	*	THIS PAGING TABLE - CORE PAGE REGION RELATIONSHIP IS RETAINED.	*
		3266	*		*
		3267	*	*****	*

S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	06/09/20	PAGE	7
				3269	*	HDR #INSTD				
				3270	*****					
				3271	*	PROGRAM HEADER FOR DISK LOAD				
				3272	*****					
			0020	3273	#\$INST EQU	X'0020'				DISK ADDR OF #INSTD
			0600	3274	\$\$\$\$INS EQU	X'0600'				CORE LOAD ADDRESS OF #INSTD
			0010	3275	#\$@INS EQU	016				SECTOR CNT OF #INSTD
0600				3276		ORG \$\$\$INS				CORE LOAD ADDRESS
			0600	3277	\$\$\$\$\$\$\$ EQU	*				FIRST LOCATION IN PROGRAM
0600	7BC9D5E2E3C4		0605	3278		DC CL6'#INSTD'				PROGRAM NAME
0606	03		0606	3279		DC IL1'003'				PROGRAM NUMBER OF #INSTD
			0607	3280	#INST EQU	*				ENTRY POINT TO PROGRAM
				3281	*** END OF EXPANSION ***					



[illegible]

## S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER	MOD	DATE	PAGE	
					3300		*****					
					3301		* ROUTINE TO UTILIZE EXTENDED CORE FOR VIRTUAL MEMORY PAGES					
					3302		*****					
					3303		*					
					3304		* ADJUST CORE PAGE REGION PARAMETERS FOR EXTENDED CORE					
					3305		*					
0612	4E	00	CF	043B	3306	IMI020	ALC IMIPCT(,@BR),\$EXFTR(1) ADJUST THE CORE PAGE COUNT					
0617	5F	00	CF	C3	3307		SLC IMIPCT(,@BR),IMIBN1(1,@BR) * FOR EXTENDED CORE REGION					
061B	5E	00	D0	C3	3308		ALC IMIPAD-1(,@BR),IMIBN1(1,@BR) ADJUST START OF CORE PAGES					
					3309		* * FOR EXPANDED PAGING TABLES					
					3310		*					
					3311		* INITIALIZE EXPANDED CORE VIRTUAL MEMORY WITH 1ST VIRTUAL MEMORY PAGES					
					3312		*					
061F	C0	87	0025		3313	IMI030	B \$DISKN LINK TO LOAD CORE PAGES					
0623	06D7			0624	3314		DC AL(@CADDR)(IMIPDP) PAGE CORELOAD DPL CADDR					
					3315		*					
					3316		* ADJUST PAGING MODULE PARAMETERS AND TABLES FOR EXPANDED NO. OF PAGES					
					3317		*					
0625	4C	00	C7	043B	3318	IMI040	MVC IMIEX1(,@BR),\$EXFTR(1) COMPUTE NO. OF ADDITIONAL CORE					
062A	5F	00	C7	C3	3319		SLC IMIEX1(,@BR),IMIBN1(1,@BR) * PAGES DUE TO THE EXTENSION					
062E	5C	00	C9	C7	3320		MVC IMIEX2(,@BR),IMIEX1(1,@BR) COMPUTE TWICE THE NUMBER OF					
0632	5E	00	C9	C7	3321		ALC IMIEX2(,@BR),IMIEX1(1,@BR) * ADDITIIONAL CORE PAGES					
0636	5C	00	CB	C9	3322		MVC IMIEX3(,@BR),IMIEX2(1,@BR) COMPUTE THRICE THE NUMBER OF					
063A	5E	00	CB	C7	3323		ALC IMIEX3(,@BR),IMIEX1(1,@BR) * ADDITIONAL CORE PAGES					
					3324		*					
063E	0E	00	12EB	043B	3325		ALC IPGSZ1+@Q,\$EXFTR(1) ADJUST SYSTEM PAGE SIZE PARAM					
0644	0E	00	1374	043B	3326		ALC IPGSZ2+@Q,\$EXFTR(1) ADJUST SYSTEM PAGE SIZE PARAM					
064A	0E	00	1424	043B	3327		ALC IPGSZ3+@Q,\$EXFTR(1) ADJUST SYSTEM PAGE SIZE PARAM					
					3328		*					
0650	1E	01	13C7	CB	3329		ALC IPGUT1+@OP1,IMIEX3(@CADDR,@BR) ADJUST PAGE USAGE TBL ADDR					
0655	1E	01	13C9	CB	3330		ALC IPGUT1+@OP2,IMIEX3(@CADDR,@BR) ADJUST PAGE USAGE TBL ADDR					
065A	1E	01	143E	C7	3331		ALC IPGUT2,IMIEX1(@CADDR,@BR) ADJUST PAGE USAGE TBL ADDR					
					3332		*					
065F	1F	01	1445	C7	3333		SLC IPGMX1,IMIEX1(@REGL,@BR) ADJUST MAX CORE PAGE CNT PARAM					
0664	1E	00	13C5	C9	3334		ALC IPGMX2+@Q,IMIEX2(1,@BR) ADJUST MAX CORE PAGE CNT PARAM					
0669	1E	00	145F	C7	3335		ALC IPGMX3+@D1,IMIEX1(1,@BR) ADJUST MAX CORE PAGE CNT PARAM					
066E	1E	00	146C	C7	3336		ALC IPGMX4+@D1,IMIEX1(1,@BR) ADJUST MAX CORE PAGE CNT PARAM					
0673	1E	00	1465	C7	3337		ALC IPGMX5+@Q,IMIEX1(1,@BR) ADJUST MAX CORE PAGE CNT PARAM					
					3338		*					
0678	0F	5F	165F	165F	3339		SLC I\$CPG1+I@LXPT-1,I\$CPG1+I@LXPT-1(I@LXPT) ZERO THE EXPANDED					
					3340		* * CORE PAGE TALE ENTRY AREAS					
067E	F2	87	06		3341		J IMI060 SKIP TO CONTINUE INITIALIZING					
					3342		*					
					3343		*****					

## S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20 PAGE 10
					3345	*****		
					3346	* CORE VIRTUAL MEMORY PAGE REGION INITIALIZATION ROUTINE		
					3347	*****		
					3348	*		
					3349	* INITIALIZE MINIMUM CORE VIRTUAL MEMORY WITH 1ST VIRTUAL MEMORY PAGES		
					3350	*		
0681	C0	87	0025		3351	IMI050 B	\$DISKN	LINK TO LOAD CORE PAGES
0685	06D7			0686	3352	DC	AL(@CADDR)(IMIPDP)	PAGE CORELOAD DPL CADDR
					3353	*		
					3354	* INITIALIZE THE PAGING ROUTINE CORE PAGE REFERENCE TABLE		
					3355	*		
0687	C2	02	14CA		3356	IMI060 LA	IZPGTB,@XR	LOAD ADDRESS OF 1ST TABLE ENTRY
068B	5C	00	85 CF		3357	MVC	IMI070+@Q(,@BR),IMIPCT(1,@BR)	SET HIGHEST CORE PAGE NO.
					3358	*		
068F	BC	00	00		3359	IMI070 MVI	I@PRTE(,@XR),*-*	SET CORE PAGE NO. IN PG REF TBL
0692	E2	02	01		3360	LA	@B1(,@XR),@XR	INCREMENT THE TABLE POINTER
0695	5F	00	85 C3		3361	SLC	IMI070+@Q(,@BR),IMIBN1(1,@BR)	DECREMENT CORE PAGE NO.
0699	D0	84	84		3362	BH	IMI070(,@BR)	REPEAT LOOP UNTIL PAGE NO. = 0
					3363	*		
					3364	* TEST SYSTEM EXECUTION INDICATOR-1 FOR FLOW TRACE PROCESSING.		
					3365	*		
069C	38	08	03D0		3366	IMI080 TBN	\$XIND1,\$TFLOW	TEST FOR FLOW TRACE EXECUTION
06A0	F2	90	04		3367	JF	IMI100	BRANCH IF NOT FLOW TRACE MODE
					3368	*		
					3369	* INITIALIZE INTERPRETER CORE RESIDENT ROUTINES FOR FLOW TRACE MODE		
					3370	*		
06A3	3C	80	0D2B		3371	IMI090 MVI	IZTFSW,@NOP	SET EXECUTIVE RTN FLOW TRACE
					3372	*		
					3373	* ESTABLISH INTERNAL PROGRAM DATA FILE POINTERS		
					3374	*		
06A7	0C	01	0D53 03CF		3375	IMI100 MVC	IZDATA,\$INLNO(@VADDR)	SET INTERNAL DATA FILE POINTER
06AD	0C	01	0D55 03CF		3376	MVC	IZDATI,\$INLNO(@VADDR)	SET DATA-FILE 1ST ELEMENT VADDR
06B3	1C	01	03CF C5		3377	MVC	\$INLNO,IMIHLN(B@LCLN,@BR)	SET DUMMY INITIAL LINE NO.
					3378	*		
					3379	* WAIT FOR COMPLETION OF CORE PAGE AREA LOADING		
					3380	*		
06B8	C0	87	0025		3381	IMM110 B	\$DISKN	LINK TO WAIT INPUT COMPLETED
06BC	057F			06BD	3382	DC	AL(@CADDR)(\$WAITF)	CADDR OF DISK IOCR 'WAIT' DPL
					3383	*		
					3384	* SET SYSTEM INDICATORS FOR EXECUTION MODE		
					3385	*		
06BE	3C	80	0476		3386	IMI120 MVI	\$CIMSK,@NOP	SET CONSOLE INTERRUPT MASK
06C2	3A	80	03D0		3387	SBN	\$XIND1,\$VMDEF	SET V.M. DEFINITION INDICATOR ON
06C6	3A	01	03D1		3388	SBN	\$XIND2,\$EXCMD	SET EXECUTION MODE INDICATOR
					3389	*		
					3390	* BRANCH TO BEGIN PSEUDO INSTRUCTION EXECUTION		
					3391	*		
06CA	C0	87	0C5C		3392	IMI130 B	INTERP	BRANCH TO EXECUTE PMC
					3393	*		
					3394	*****		

## S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 11
			3396		*****	
			3397		* INTERPRETER INITIATOR PROGRAM CONSTANTS	
			3398		*****	
			3399		*	
06CE	01		06CE	3400	IMIBN1 DC IL1'1' BINARY INTEGER .1	
06CF	FFFF		06D0	3401	IMIHLN DC XL(B@LCLN)'FFFF' DUMMY STARTING LINE NUMBER	
			3402		*	
			3403		*****	
			3404		* INTERPRETER INITIATOR PROGRAM WORK AREA	
			3405		*****	
			3406		*	
06D1	00		06D1	3407	DC XL1'00' ZERO FILLER FOR 2-BYTE FIELD	
06D2			06D2	3408	IMIEX1 DS CL1 1*(NO. OF EXTRA CORE PAGES)	
			3409		*	
06D3	00		06D3	3410	DC XL1'00' ZERO FILLER FOR 2-BYTE FIELD	
06D4			06D4	3411	IMIEX2 DS CL1 2*(NO. OF EXTRA CORE PAGES)	
			3412		*	
06D5	00		06D5	3413	DC XL1'00' ZERO FILLER FOR 2-BYTE FIELD	
06D6			06D6	3414	IMIEX3 DS CL1 3*(NO. OF EXTRA CORE PAGES)	
			3415		*	
			3416		*****	
			3417		* INTERPRETER INITIATOR DISK PARAMETER LISTS	
			3418		*****	
			3419		*	
			06D7	3420	IMIPDP EQU * CORE PAGE INITIALIZATION DPL	
			3421		*	
06D7	01		06D7	3422	IMIPDF DC AL1(@DGET) DISK IOCR 'READ' FUNC CODE	
06D8	07		06D8	3423	IMIPDC DC AL1(@DVBCY) PHYSICAL DISK ADDRESS FOR	
06D9	00		06D9	3424	IMIPDS DC XL1'00' * 1ST PAGE IN VIRTUAL MEMORY	
			3425		*	
06DA			06DA	3426	IMIPCT DS CL1 SECTOR COUNT FOR CORE PAGES	
06DA				3427	ORG *-1 INITIALIZE CORE PAGE SECTOR	
06DA	0A		06DA	3428	DC AL1(I@NCPG) * COUNT FOR MINIMUM CORE (8K)	
			3429		*	
06DB			06DC	3430	IMIPAD DS CL(@CADDR) CORE PAGE REGION STARTING CADDR	
06DB				3431	ORG *-@CADDR INITIALIZE CORE PAGE REGION FOR	
06DB	1600		06DC	3432	DC AL(@CADDR)(I\$CPG1) * MINIMUM CORE SIZE SYSTEM (8K)	
			3433		*	
			3434		*****	
			3435		*	
			3436		* END OF INTERPRETER INITIATOR CODING *****	

S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20	PAGE 12
				3438	*****			
				3439	* INTERPRETER FUNCTION WORK AREA AND RUN-TIME STACK			
				3440	*****			
0607				3441	ORG IMINIT			RESET TO OVERLAY THE INITIATOR
				3442	*			
			0607	3443	IMIWRK EQU *			FUNCTION WORK AREA BASE MDR
0607			0638	3444	DS CL(3*I@LUFL+2)			DEFINE THE FUNCTION WORK AREA
				3445	* IMINIT			
			0639	3446	IMISTB EQU *			RUN-TIME STACK BASE CORE ADDR
0639			0728	3447	DS CL240			DEFINE THE RUN-TIME STACK
				3448	*			
				3449	*****			
				3451	* PATCH 40			PATCH AREA 1-4
				3452	*****			
				3453	* PATCH AREA 1			
				3454	*****			
0729			0750	3455	\$\$\$\$\$1 DS CL40			PATCH AREA FOR PROGRAM
				3456	*** END OF EXPANSION ***			

## S/3 BASIC INTERPRETER - ADD/SUBTRACT RTN PROLOGUE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20	PAGE 13
		3458		*****			
		3459	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
		3460	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
		3461	*				*
		3462		*****			*
		3463	*	STATUS			*
		3464	*	VERSION 1 MODIFICATION 0			*
		3465	*				*
		3466	*	FUNCTION			*
		3467	*	* FDIADD PERFORMS A FLOATING POINT ADDITION ON THE FIRST TWO			*
		3468	*	STACK ELEMENTS.			*
		3469	*	* ENTRY FDISUB SUBTRACTS THE SECOND ELEMENT FROM THE FIRST BY			*
		3470	*	CHANGING THE SIGN OF THE SECOND ELEMENT, AND THEN PASSING			*
		3471	*	CONTROL TO FDIADD.			*
		3472	*	* FDIADD RETURNS THE SUM IN THE FIRST STACK ELEMENT IN UNPACKED			*
		3473	*	FLOATING POINT FORMAT.			*
		3474	*				*
		3475	*	ENTRY POINTS			*
		3476	*	* THE ENTRY TO ADD IS FDIADD. THE FORMAT OF THE CALLING SEQUENCE			*
		3477	*	IS AS FOLLOWS:			*
		3478	*	B I\$FADD			*
		3479	*	* THE ENTRY TO SUBTRACT IS FDISUB. THE FORMAT OF THE CALLING			*
		3480	*	SEQUENCE IS AS FOLLOWS:			*
		3481	*	B I\$FSUB			*
		3482	*				*
		3483	*	INPUT			*
		3484	*	* THE INPUT IS TWO FLOATING POINT NUMBERS, IN THE FIRST TWO STACK			*
		3485	*	ELEMENTS, TO BE ADDED OR SUBTRACTED.			*
		3486	*	* THE ADDRESS RECALL REGISTER (ARR) IS STORED, AND CONTROL IS			*
		3487	*	RETURNED BY BRANCHING TO ITS ADDRESS.			*
		3488	*				*
		3489	*	OUTPUT			*
		3490	*	* THE RESULT IS LEFT IN THE FIRST ELEMENT OF THE STACK, IN			*
		3491	*	UNPACKED FLOATING, POINT FORMAT.			*
		3492	*	* IN THE EVENT OF AN ERROR, THE APPROPRIATE CODE IS PLACED IN THE			*
		3493	*	INTERPRETER ONE-BYTE ERROR LOCATION IZERRC.			*
		3494	*				*
		3495	*	EXTERNAL REFERENCE			*
		3496	*	INTERPRETER STACK - FIRST TWO ELEMENTS			*
		3497	*	IZSTAK - LOCATION OF THE ADDRESS OF THE INTERPRETER STACK			*
		3498	*	IZERRC - ONE-BYTE INTERPRETER ERROR LOCATION			*
		3499	*	IZFWRK - 9(17) BYTES OF THIS WORK AREA			*
		3500	*				*
		3501	*	EXITS, NORMAL			*
		3502	*	* EXIT IS BY BRANCHING TO THE RETURN ADDRESS IN THE ADDRESS			*
		3503	*	RECALL REGISTER (ARR) STORED AT ENTRY.			*
		3504	*	* INDEX REGISTER 1 (@BR) IS RESTORED BEFORE RETURNING.			*
		3505	*	* THE RESULT IS IN THE FIRST INTERPRETER STACK ELEMENT.			*
		3506	*				*
		3507	*	EXITS, ERROR			*
		3508	*	* AN ERROR CODE IS PLACED IN THE 1-BYTE INTERPRETER AREA, IZERRC			*
		3509	*	* EXIT IS BY BRANCHING TO THE RETURN ADDRESS OF THE STORED ADDRESS			*
		3510	*	RECALL REGISTER (ARR).			*
		3511	*				*
		3512	*	TABLES/WORK AREA			*
		3513	*	* THE CONSTANTS & WORK AREAS INSIDE AT THE END OR THE EXECUTABLE			*

## S/3 BASIC INTERPRETER - ADD/SUBTRACT RTN PROLOGUE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 14
		3514	*	CODE AND ARE REFERENCED BY A DISPLACEMENT RELATIVE TO THE VALUE	*
		3515	*	IN INDEX REGISTER 1 (@BR).	*
		3516	*		*
		3517	*	*ATTRIBUTES	*
		3518	*	REUSABLE, RELOCATABLE.	*
		3519	*		*
		3520	*	*CHARACTER CODE DEPENDENCY	*
		3521	*	THE OPERATION OF THIS MODULE DEPENDS UPON A ZONED DECIMAL DIGIT	*
		3522	*	BEING REPRESENTED WITH THE ZONE (FIRST FOUR BITS) BEING AN -F-	*
		3523	*	FOR POSITIVE AND A -D- FOR NEGATIVE. THE DECIMAL NUMBERS MUST	*
		3524	*	BE CODED SO THAT THE LOW ORDER FOUR BITS, WHEN CONSIDERED AS A	*
		3525	*	BINARY INTEGER, IDENTIFY THE VALUE OF THE DIGIT.	*
		3526	*	THESE PROPERTIES ARE USED AT THE ENTRY FDISUB, AND IN THE	*
		3527	*	INSTRUCTIONS FOLLOWING FDI300.	*
		3528	*		*
		3529	*	*NOTES	*
		3530	*	ERROR PROCEDURES	*
		3531	*	THE ERROR CODE IS SET, AND CONTROL RETURNED TO THE CALLING	*
		3532	*	PROGRAM.	*
		3533	*		*
		3534	*	REGISTER USAGE	*
		3535	*	INDEX REGISTER 1 (@BR) IS SAVED AND RESTORED. THIS REGISTER	*
		3536	*	IS USED AS THE BASE REGISTER DURING EXECUTION.	*
		3537	*	INDEX REGISTER 2 (@XR) IS LOADED TO CONTAIN THE ADDRESS OR THE	*
		3538	*	FIRST BYTE OF THE INTERPRETER STACK, TO REFERENCE THE STACK.	*
		3539	*	@XR IS NOT SAVED OR RESTORED.	*
		3540	*		*
		3541	*	SAVED/RESTORED AREAS	*
		3542	*	NONE.	*
		3543	*		*
		3544	*	MODIFICATION CONSIDERATIONS	*
		3545	*	FDIADD MAY NOT USE ANY FURTHER INTERPRETER STACK OR WORK AREA	*
		3546	*	WITHOUT AFFECTING THE OTHER MATHEMATIC FUNCTION ROUTINES.	*
		3547	*	THE ZONED DECIMAL INSTRUCTIONS HANDLE THE LARGEST OPERANDS	*
		3548	*	POSSIBLE IN LONG PRECISION. THEREFORE, THERE CAN BE NO	*
		3549	*	GREATER ACCURACY OR SIGNIFICANCE.	*
		3550	*		*
		3551	*	REQUIRED MODULES	*
		3552	*	@SYSEQ - COMMON SYSTEM EQUATES	*
		3553	*	@ERMEQ - ERROR MESSAGE EQUATES	*
		3554	*	\$B@EQU - COMPILER SYSTEM EQUATES	*
		3555	*	IZCOMN - CORE RESIDENT COMMON LOCATION EQUATES	*
		3556	*	\$I\$SET - STANDARD PRECISION EXEUTION EQUATES	*
		3557	*		*
		3558	*	OTHER	*
		3559	*	NONE	*
		3560	*	*****	*



## S/3 BASIC INTERPRETER - ADD/SUBTRACT RTN PROLOGUE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 15
				3562	*****	
				3563	* FDIADD S/3 ADD/SUBTRACT FLOATING POINT ROUTINE	
				3564	*****	
				3565	*	
			075D	3566	USING FDIADD,@BR	
				3567	*	
				3568	* EXECUTION TO SUBTRACT ROUTINE FOLLOWS	
				3569	*	
0751	35	02 0D4E		3570	FDISUB L IZSTAK,@XR LOAD STACK POINTER	
				3571	*	
				3572	* THE FOLLOWING TWO INSTRUCTIONS REVERSE THE SIGN OF B	
				3573	*	
0755	8E	00 0F 081B		3574	ALC I@RSE2(1,@XR),FDIPL1 INCREMENT SIGN ZONE BY 0001	
075A	BA	D0 0F		3575	SBN I@RSE2(,@XR),B@ZNEG CHANGE SIGN	
				3576	*	
				3577	* EXECUTION ENTRY TO ADD ROUTINE FOLLOWS	
				3578	*	
075D	34	01 0813		3579	FDIADD ST FDI888+@OP1,@BR SAVE @BR	
0761	C2	01 075D		3580	LA FDIADD,@BR NEW BASE ADDRESS	
0765	74	08 BA		3581	ST FDI890+@OP1(,@BR),@ARR RETURN ADDRESS	
0768	35	02 0D4E		3582	L IZSTAK,@XR STACK POINTER	
				3583	*	
				3584	* THE INSTRUCTION AT FDI300, WHICH ADDS A AND B. IS MODIFIED SO AS TO	
				3585	* PROVIDE FOR ALIGNMENT OR DECIMAL POINTS OF A AND B. AND SO MUST BE	
				3586	* RESTORED BEFORE EXECUTION EACH TIME THE ROUTINE IS ENTERED.	
				3587	*	
076C	5C	02 5A C2		3588	FDI010 MVC FDI300+@DD2(FDIINT,@BR),FDIINI+@DD2(,@BR) INITIALIZE INST	
0770	6C	00 9D 00		3589	MVC FDI320+@Q(1,@BR),I@1SE1+I@DEXP(,@XR) SAVE EXP OF A	
0774	AD	00 00 08		3590	CLC I@1SE1+I@DEXP(1,@XR),I@1SE2+I@DEXP(,@XR) COMPARE EXPS A:B	
				3591	*	
				3592	* IF FLOATING POINT EXPONENTS OF A AND B ARE THE SAME, A AND B CAN BE	
				3593	* ADDED IMMEDIATELY WITHOUT MODIFYING THE ADD INSTRUCTION.	
				3594	* IF THE EXPONENT OF A EXCEEDS THE EXPONENT OF B, THE MANTISSA OF B	
				3595	* MUST BE SHIFTED RIGHT BEFORE ADDING IT TO THE MANTISSA OF A.	
				3596	* IF THE EXPONENT OF B EXCEEDS THE EXPONENT OF A. THE NUMBERS A AND B	
				3597	* ARE INTERCHANGED TO REDUCE TO THE PRECEDING CASE.	
				3598	*	
0778	D0	81 51		3599	BE FDI299(,@BR) DECIMAL POINT ALREADY ALIGNED	
077B	D0	84 33		3600	BH FDI230(,@BR) EXP OF A EXCEEDS EXP OF B	
077E	2C	07 060F 07		3601	MVC IZFWRK+I@LUFV(I@LUFV),I@RSE1(,@XR) SAVE A	
0783	AC	07 07 0F		3602	MVC I@RSE1(I@LUFV,@XR),I@RSE2(,@XR) INTERCHANGE B	
0787	8C	07 0F 060F		3603	MVC I@RSE2(I@LUFV,@XR),IZFWRK+I@LUFV AND A	
078C	6C	00 9D 00		3604	MVC FDI320+@Q(1,@BR),I@1SE1+I@DEXP(,@XR) SAVE EXP OF NEW A	
0790	AF	00 00 08		3605	FDI230 SLC I@1SE1+I@DEXP(1,@XR),I@1SE2+I@DEXP(,@XR) N-EXP(A)-EXP(8)	
0794	BD	07 00		3606	CLI I@1SE1+I@DEXP(,@XR),I@PREC COMPARE N WITH THE PRECISION	
				3607	*	
				3608	* IF THE EXPONENTS DIFFER BY I@PREC OR MORE, THE SMALLER NUMBER (WHICH	
				3609	* IS ALWAYS B AT THIS POINT) IS DROPPED, LEAVING A AS THE RESULT.	
				3610	*	
0797	D0	84 9C		3611	BH FDI320(,@BR) IN THE CASE A + B = A	
				3612	*	
				3613	* MODIFY THE INSTRUCTION AT FDI300, WHICH ADDS A AND B, SO THAT THE	
				3614	* DECIMAL POINTS ARE ALIGNED DURING THE ADDITION. TO DO THIS, THE	
				3615	* DISPLACEMENT TO B IS DECREMENTED BY N = EXP(A) - EXP(B), AND THE	
				3616	* LENGTH CODE FOR B IS ALSO DECREMENTED BY N. THE MACHINE LANGUAGE	
				3617	* 4 BIT LENGTH CODE FOR A MUST BE INCREASED BY N TO KEEP THE LENGTH	



## S/3 BASIC INTERPRETER - ADD/SUBTRACT RTN PROLOGUE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 06/09/20 PAGE 16

```

3618 * OF THE SAME, THE NUMBER N IS ALWAYS POSTIVE AND LESS THAN I@PREC.
3619 *
079A 6F 00 5A 00 3620 SLC FDI300+@DD2(1,@BR),I@1SE1+I@DEXP(,@XR) DECREMENT DISPLCMN
079E 6F 00 58 00 3621 SLC FDI300+@Q(1,@BR),I@1SE1+I@DEXP(,@XR) DECREMENT L2
07A2 68 01 58 00 3622 MZN FDI300+@Q(,@BR),I@1SE1+I@DEXP(,@XR) ADJUST L1
3623 *
3624 * MOVE SIGN ZONE OR B TO ITS NEW LOCATION IN THE RIGHTMOST BYTE OF
3625 * THE NEW (TRUNCATED) B. THE DISPLACEMENT TO THIS BYTE IS MOST
3626 * CONVENIENT OBTAINED FROM THE MODIFIED SECOND OPERAND OF THE
3627 * INSTRUCTION AT FDI300. FINALLY, INSERT LEADING DECIMAL ZEROES
3628 * IN THE EXPONENT PARTS OR A AND B.
3629 *
07A6 5C 00 4F 5A 3630 MVC FDI285+@D1(1,@BR),FDI300+@DD2(,@BR) SAVE DISPLACEMENT TO
07AA A8 00 00 0F 3631 FDI285 MZZ *-*(,@XR),I@RSE2(,@XR) INSERT SIGN OF B
07AE BC F0 00 3632 FDI299 MVI I@1SE1+I@DEXP(,@XR),@DZERO INSERT LEADING ZERO
07B1 BC F0 08 3633 MVI I@1SE2+I@DEXP(,@XR),@DZERO INSERT LEADING ZERO
3634 *
3635 * ADD A AND B. THE FOLLOWING INSTRUCTION HAS BEEN MODIFIED INTO --
3636 * AZ I@RSE1(I@LUFV,@XR),I@RSE2-N(I@LUFV,@XR)
3637 * WHERE N = EXP(A) - EXP(B)
3638 *
07B4 A6 00 07 00 3639 FDI300 AZ I@RSE1(@VQ,@XR),*-*(@VQ,@XR) ADD A AND B
3640 *
3641 * NORMALIZE THE RESULT
3642 *
07B8 68 00 BC 07 3643 MZZ FDISGN(,@BR),I@RSE1(,@XR) SAVE SIGN OF RESULT
07BC BA F0 07 3644 SBN I@RSE1(,@XR),B@ZPOS FORCE SIGN POSITIVE
07BF A7 06 0F 0F 3645 SZ I@RSE2(I@PREC,@XR),I@RSE2(I@PREC,@XR) CLEAR TO DECIMAL 0'
3646 *
3647 * INITIALIZE COUNTER TO -1. STARTING WITH THE LEFTMOST BYTE OF THE
3648 * RESULT, THE LEADING ZEROES WILL BE COUNTED. IF THE FIRST DIGIT IS
3649 * NOT ZERO, AN OVERFLOW INTO THE EXPONENT BYTE (INITIALLY CLEARED TO
3650 * ZERO) HAS OCCURED AND THE EXPONENT OF THE RESULT SET TO THE ORIGINAL
3651 * EXPONENT OF 'A' PLUS THE CONTENTS OF THE COUNTER (-1 AT THIS POINT).
3652 * THE COUNTER IS INCREMENTED BY ONE FOR EACH LEADING ZERO UNTIL THE
3653 * FIRST NON-ZERO DIGIT IS DETECTED.
3654 *
07C3 7C FF BB 3655 MVI FDICTR(,@BR),X'FF' INITIALIZE COUNTER TO -1
07C6 B9 0F 00 3656 FDI310 TBF I@1SE1(,@XR),X'0F' IS THE DIGIT A ZERO ?
07C9 D0 90 86 3657 BF FDI315(,@BR) NO
07CC E2 02 01 3658 LA FDIINC(,@XR),@XR INCREMENT @XR
07CF 5E 00 BB BD 3659 ALC FDICTR(1,@BR),FDIICR(,@BR) INCREMENT COUNTER BY ONE
07D3 7D 07 BB 3660 CLI FDICTR(,@BR),I@PREC WAS THIS LAST DIGIT ?
07D6 D0 01 69 3661 BNE FDI310(,@BR) NO, REPEAT
3662 *
3663 * ALL DIGITS OF RESULT ARE ZEROES, SO PUT FLOATING POINT ZERO IN THE
3664 * STACK AND RETURN TO CALLING PROGRAM, THE MANTISSA IS ALREADY ALL
3665 * ZEROES AT THIS POINT.
3666 *
07D9 35 02 0D4E 3667 L IZSTAK,@XR RESTORE XR
07DD BC 1E 00 3668 MVI I@1SE1+I@DEXP(,@XR),B@NXLO SET EXPONENT TO -98
07E0 D0 87 B3 3669 B FDI888(,@BR) EXIT
3670 *
3671 * MOVE RESULT INTO STACK VIA TEMPORARY LOCATION I$FWRK, EXPONENT OF
3672 * RESULT IS EXP(A)-COUNT(RESULT OF NOMALIZATION). RIGHTMOST DIGIT
3673 * OF NORMALIZED RESULT IS AT I@RSE1-1(,@XR) (WITH THE INDEX REGISTER

```

## S/3 BASIC INTERPRETER - ADD/SUBTRACT RTN PROLOGUE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 17
				3674	* STILL INCREMENTED).	
				3675	*	
	07E3	5F 00 9D BB		3676	FDI315 SLC FDI320+@Q(1,@BR),FDICTR(,@BR) INCREMENT OR DECREMENT EXP	
	07E7	2C 06 060F 06		3677	MVC IZFWRK+I@LUFV(I@PREC),I@RSE1-FDISHF(,@XR) SHIFT MANTISSA	
	07EC	35 02 0D4E		3678	L IZSTAK,@XR RESTORE XR	
	07F0	8C 06 07 060F		3679	MVC I@RSE1(I@PREC,@XR),IZFWRK+I@LUFV PUT MANTISSA IN STACK	
	07F5	98 00 07 BC		3680	MZZ I@RSE1(,@XR),FDISGN(,@BR) INSERT CORRECT SIGN	
	07F9	BC 00 00		3681	FDI320 MVI I@1SE1+I@DEXP(,@XR),*-* INSERT EXPONENT	
				3682	*	
				3683	* CHECK FOR OVERFLOW OR UNDERFLOW	
				3684	*	
	07FC	BD E3 00		3685	CLI I@1SE1+I@DEXP(,@XR),B@NXHI IS EXPONENT +99 OR MORE ?	
	07FF	D0 04 A9		3686	BNH FDI881(,@BR) NO, OVERFLOW HAS NOT OCCURED	
	0802	3C ED 0CBC		3687	MVI IZERRC,@@E791 OVERFLOW FLAG	
	0806	BD 1E 00		3688	FDI881 CLI I@1SE1+I@DEXP(,@XR),B@NXLO IS EXPONENT -98 OR LESS ?	
	0809	D0 02 B3		3689	BNL FDI888(,@BR) NO, UNDERFLOW HAS NOT OCCURED	
	080C	3C EE 0CBC		3690	MVI IZERRC,@@E792 UNDERFLOW FLAG	
	0810	C2 01 0000		3691	FDI888 LA *-*,@BR RESTORE BASE REG	
	0814	C0 87 0000		3692	FDI890 B *-* RETURN	
				3693	*	
				3694	* CONSTANTS FOR FDIADD FOLLOW	
				3695	*	
			0001	3696	FDIINC EQU 1 INCREMENT @XR TO SEARCH FOR 0'S	
			0001	3697	FDISHF EQU 1 CONSTANT IN SHIFTING MANTISSA	
			0003	3698	FDIINT EQU 3 LENGTH FOR ADD MIR INIT	
				3699	*	
				3700	* WORK AREA FOR FDIADD FOLLOWS	
				3701	*	
	0818		0818	3702	FDICTR DS CL1 COUNTER DURING NORMALIZATION	
	0819		0819	3703	FDISGN DS CL1 FOR SAVING SIGN ZONE	
				3704	*	
				3705	* CONSTANTS FOR FDIADD FOLLOW	
				3706	*	
	081A	01	081A	3707	FDIICR DC AL1(FDIONE) INCREMENT OF ONE	
	081B	10	081B	3708	FDIPL1 DC AL1(FDIZN1) SIGN ZONE INCREMENT OF ONE	
	081C	A6 07 07 0F		3709	FDIINI AZ I@RSE1(I@LUFV,@XR),I@RSE2(I@LUFV,@XR) INITIAL INSTRUCTION	
				3710	*	
			0001	3711	FDIONE EQU 1	
			0010	3712	FDIZN1 EQU X'10'	
				3713	*	
				3714	* END OF FDIADD CODING	
				3715	*	

## S/3 BASIC INTERPRETER - FLOATING MULTIPLY

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 18
		3717		*****	
		3718	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		3719	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
		3720	*		*
		3721		*****	
		3722	*	*STATUS	*
		3723	*	VERSION 1 MODIFICATION 0	*
		3724	*		*
		3725	*	*FUNCTION	*
		3726	*	* FZIMPY PERFORMS A FLOATING POINT MULTIPLICATION ON THE FIRST	*
		3727	*	TWO ELEMENTS OF THE INTERPRETER STACK.	*
		3728	*	* FZIMPY MULTIPLIES BY SUCCESSIVE ADDITIONS.	*
		3729	*	* THE PRODUCT IS LEFT IN UNPACKED FLOATING POINT FORM, IN THE	*
		3730	*	FIRST STACK ELEMENT.	*
		3731	*		*
		3732	*	*ENTRY POINTS	*
		3733	*	* THE ENTRY IS FZIMPY. THE FORMAT OF THE CALLING SEQUENCE IS AS	*
		3734	*	FOLLOWS:	*
		3735	*	B I\$FMPY	*
		3736	*		*
		3737	*	*INPUT	*
		3738	*	* THE INPUT IS TWO UNPACKED FLOATING POINT NUMBERS, IN THE FIRST	*
		3739	*	TWO INTERPRETER STACK ELEMENTS.	*
		3740	*	* THE ADDRESS RECALL REGISTER (ARR) IS STORED, AND CONTROL IS	*
		3741	*	RETURNED BY BRANCHING TO THE ADDRESS IN IT.	*
		3742	*		*
		3743	*	*OUTPUT	*
		3744	*	* THE PRODUCT IS LEFT IN THE FIRST STACK ELEMENT, IN UNPACKED	*
		3745	*	FLOATING POINT FORMAT.	*
		3746	*	* IN THE EVENT OF AN ERROR (OVERFLOW OR UNDERFLOW), THE APPRO-	*
		3747	*	PRIATE CODE IS PLACED IN THE INTERPRETER ONE-BYTE ERROR LOCATION	*
		3748	*	IZERRC.	*
		3749	*		*
		3750	*	*EXTERNAL REFERENCES	*
		3751	*	IZSTAK - LOCATION OF ADDRESS OF THE INTERPRETER STACK	*
		3752	*	INTERPRETER STACK - FIRST THREE ELEMENTS	*
		3753	*	IZERRC - ONE-BYTE INTERPRETER ERROR LOCATION	*
		3754	*		*
		3755	*	*EXITS, NORMAL	*
		3756	*	* EXIT IS BY BRANCHING TO THE RETURN ADDRESS IN THE ADDRESS	*
		3757	*	RECALL REGISTER (ARR) STORED AT ENTRY.	*
		3758	*	* INDEX REGISTER 1 (@BR) IS RESTORED BEFORE RETURNING.	*
		3759	*	* THE PRODUCT IS PLACED IN THE FIRST STACK ELEMENT REFERENCED BY	*
		3760	*	THE ADDRESS IN IZSTAK.	*
		3761	*		*
		3762	*	*EXITS, ERROR	*
		3763	*	* AN ERROR CODE IS PLACED IN THE INTERPRETER ERROR AREA, IZERRC.	*
		3764	*	* EXIT IS BY BRANCHING TO THE RETURN ADDRESS OF THE STORED ARR.	*
		3765	*		*
		3766	*	*TABLE/WORK AREA	*
		3767	*	* A JUMP TABLE IS LOCATED AT THE BEGINNING OF THE MODULE, AND IS	*
		3768	*	USED TO CONTROL THE NUMBER OF ADDITIONS TO BE PERFORMED.	*
		3769	*	* THE CONSTANTS AND WORK AREAS RESIDE AT THE END OF THE EXECUTABLE	*
		3770	*	CODE, AND ARE REFERENCED BY A DISPLACEMENT RELATIVE TO THE	*
		3771	*	VALUE IN @BR.	*
		3772	*		*

## S/3 BASIC INTERPRETER - FLOATING MULTIPLY

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20	PAGE 19
		3773	*	ATTRIBUTES			*
		3774	*	REUSABLE, RELOCATABLE.			*
		3775	*				*
		3776	*	CHARACTER CODE DEPENDENCY			*
		3777	*	THE OPERATION OF THIS MODULE DEPENDS UPON A ZONED DECIMAL DIGIT			*
		3778	*	BEING REPRESENTED WITH THE ZONE (FIRST FOUR BITS) BEING AN 'F'			*
		3779	*	FOR POSITIVE, AND A 'D' FOR NEGATIVE. THE DECIMAL NUMBERS MUST			*
		3780	*	BE CODED SO THAT THE LOW ORDER FOUR BITS, WHEN CONSIDERED AS A			*
		3781	*	BINARY INTEGER, IDENTIFY THE VALUE OF THE DIGIT.			*
		3782	*	THESE PROPERTIES ARE USED AT FZI002, AND FZI005.			*
		3783	*				*
		3784	*	NOTES			*
		3785	*	ERROR PROCEDURES			*
		3786	*	THE ERROR CODE IS SET, AND CONTROL RETURNED TO THE CALLING			*
		3787	*	PROGRAM.			*
		3788	*				*
		3789	*	REGISTER USAGE			*
		3790	*	INDEX REGISTER 1 (@BR) IS SAVED AND RESTORED. THIS REGISTER			*
		3791	*	IS USED AS THE BASE REGISTER DURING EXECUTION.			*
		3792	*	INDEX REGISTER 2 (@XR) IS LOADED TO CONTAIN THE ADDRESS OF THE			*
		3793	*	FIRST BYTE OF THE INTERPRETER STACK, TO REFERENCE THE STACK.			*
		3794	*	@XR IS NOT SAVED OR RESTORED.			*
		3795	*				*
		3796	*	SAVED/RESTORED AREAS			*
		3797	*	NONE.			*
		3798	*				*
		3799	*	MODIFICATION CONSIDERATIONS			*
		3800	*	NONE.			*
		3801	*				*
		3802	*	REQUIRED MODULES			*
		3803	*	@SYSEQ - COMMON SYSTEM EQUATES			*
		3804	*	@ERMEQ - ERROR MESSAGE EQUATES			*
		3805	*	\$B@EQU - COMPILER SYSTEM EQUATES			*
		3806	*	IZCOMN - CORE RESIDENT COMMON LOCATION EQUATES			*
		3807	*	\$I@SEQ - STANDARD PRECISION EXECUTION EQUATES			*
		3808	*				*
		3809	*	OTHER			*
		3810	*	NONE.			*
		3811	*	*****			*

## S/3 BASIC INTERPRETER - FLOATING MULTIPLY

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 06/09/20 PAGE 20

```

3813 *****
3814 * FZIMPY MULTIPLY FLOATING POINT ROUTINE
3815 *****
3816 *
0820 3817 USING *,@BR BASE REG POINTS HERE ON ENTRY
3818 *
3819 * THE FOLLOWING TABLE MUST BEGIN AT RELATIVE DISPLACEMENT ZERO.
3820 * THIS IS THE JUMP TABLE, CONSISTING OF 10 ONE BYTE DISPLACEMENTS,
3821 * WHICH CORRESPOND TO THE DIGITS 0 THROUGH 9. THE MULTIPLIER DIGITS,
3822 * STARTING FROM THE RIGHT, ARE USED TO INDEX THIS TABLE TO GET THE
3823 * CORRECT DISPLACEMENT TO JUMP INTO THE ADD ROUTINE.
3824 *
0820 2B 0820 3825 FZITAB DC XL1'2B' MUST BE AT TOP OF PAGE
0821 100C0804 0824 3826 DC XL4'100C0804'
0825 00171B1F23 0829 3827 DC XL5'00171B1F23'
3828 *
3829 * EXECUTION ENTRY FOR FZIMPY FOLLOWS
3830 *
082A 34 01 08F7 3831 FZIMPY ST FZI888+@OP1,@BR SAVE BASE REG
082E C2 01 0820 3832 LA FZITAB,@BR LOAD BASE REG TO JUMP TABLE
0832 74 08 DB 3833 ST FZI890+@OP1(,@BR),@ARR SAVE RETURN ADDRESS
0835 35 02 0D4E 3834 L IZSTAK,@XR LOAD STACK POINTER
3835 *
3836 * EXECUTION ENTRY FOR FZIMPY FOLLOWS
3837 *
0839 6C 07 E6 0F 3838 MVC FZIMUC(I@LUFV,@BR),I@RSE2(,@XR) SAVE MULTIPLICAND (B)
3839 *
3840 * COMPARE SIGNS OF A AND B AND SAVE CONDITION REGISTER SO THAT THE
3841 * RESULT OF THIS COMPARE CAN BE USED LATER TO DETERMINE THE SIGN OF
3842 * THE PRODUCT A+B. MEANWHILE, MAKE B (THE MULTIPLICAND) POSITIVE.
3843 * A (THE MULTIPLIER) NEED NOT BE POSITIVE BECAUSE ONLY THE NUMERIC
3844 * PARTS OF THE ZONED DECIMAL DIGITS ARE USED TO INDEX THE JUMP TABLE.
3845 *
083D A8 03 0F 07 3846 MNN I@RSE2(,@XR),I@RSE1(,@XR) SAVE NUMERIC OF B TO THAT OF A
0841 AD 00 0F 07 3847 CLC I@RSE2(1,@XR),I@RSE1(,@XR) COMPARE SIGNS
0845 74 04 DD 3848 ST FZISUM(,@BR),@PSR SAVE CONDITION REG
0848 7A F0 E6 3849 FZI002 SBN FZIMUC(,@BR),B@ZPOS FORCE MULTIPLICAND POSITIVE
084B BC F0 17 3850 MVI I@RSE3(,@XR),@DZERO GET DECIMAL ZERO
084E AC 0E 16 17 3851 MVC I@RSE3-1(I@LUFV+I@PREC,@XR),I@RSE3(,@XR) CLEAR ACCUMULATO
0852 7C 07 DE 3852 MVI FZICTR(,@BR),I@PREC SET LOOP COUNTER
3853 *
3854 * COMPUTE ENTRY POINT TO ADD ROUTINE AS FOLLOWS -
3855 * * INSERT NUMERIC PART OF MULTIPLIER DIGIT INTO THE SECOND
3856 * DISPLACEMENT FIELD OF A MVC INSTRUCTION. THE FIRST 4 BITS OF
3857 * THIS FIELD ARE ALWAYS 0000 SINCE *-* IS USED FOR THE FIELD DURING
3858 * ASSEMBLY. THE DISPLACEMENT POINTS TO THE CORRECT BYTE IN THE
3859 * JUMP TABLE.
3860 * * THE MVC INSTRUCTION THEN MOVES THE CORRESPONDING JUMP DISPLACEMENT
3861 * FROM THE JUMP TABLE INTO THE DISPLACEMENT FIELD OF A JUMP INSTRU-
3862 * TION, WHICH WILL ENTER THE ADD ROUTINE AT THE CORRECT POINT.
3863 *
0855 68 03 3C 07 3864 FZI009 MNN FZI010+@DD2(,@BR),I@RSE1(,@XR) USE DIGIT TO IDX JUMP TABLE
0859 5C 00 3F 00 3865 FZI010 MVC FZI020+@D1(1,@BR),*-(,@BR) GET CORRESPONDING DISPLACEMNT
085D F2 87 00 3866 FZI020 J *-* ENTER ADD ROUTINE AS PER DIGIT
3867 *
3868 * ADD ROUTINE

```



## S/3 BASIC INTERPRETER - FLOATING MULTIPLY

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER	15,	MOD	00	06/09/20	PAGE	21
					3869	*								
	0860	96	16 17 E6		3870	AZ	I@RSE3(I@LUFV,@XR),FZIMUC(I@PREC,@BR)	ENTRY FOR 5						
	0864	96	16 17 E6		3871	AZ	I@RSE3(I@LUFV,@XR),FZIMUC(I@PREC,@BR)	ENTRY FOR 4						
	0868	96	16 17 E6		3872	AZ	I@RSE3(I@LUFV,@XR),FZIMUC(I@PREC,@BR)	ENTRY FOR 3						
	086C	96	16 17 E6		3873	AZ	I@RSE3(I@LUFV,@XR),FZIMUC(I@PREC,@BR)	ENTRY FOR 2						
	0870	96	16 17 E6		3874	AZ	I@RSE3(I@LUFV,@XR),FZIMUC(I@PREC,@BR)	ENTRY FOR 1						
	0874	D0	87 6B		3875	B	FZI030(,@BR)	EXIT LOOP						
	0877	97	16 17 E6		3876	SZ	I@RSE3(I@LUFV,@XR),FZIMUC(I@PREC,@BR)	ENTRY FOR 6						
	087B	97	16 17 E6		3877	SZ	I@RSE3(I@LUFV,@XR),FZIMUC(I@PREC,@BR)	ENTRY FOR 7						
	087F	97	16 17 E6		3878	SZ	I@RSE3(I@LUFV,@XR),FZIMUC(I@PREC,@BR)	ENTRY FOR 8						
	0883	97	16 17 E6		3879	SZ	I@RSE3(I@LUFV,@XR),FZIMUC(I@PREC,@BR)	ENTRY FOR 9						
	0887	96	07 17 E7		3880	AZ	I@RSE3(I@LUFV,@XR),FZIMUC+FZIONE(I@LUFV,@BR)	RECOMPLEMENT						
					3881	*								
	088B	76	02 EA		3882	FZI030 A	FZIMI1(,@BR),@XR	DECREMENT POINTER TO NEXT DIGIT						
	088E	5E	00 DE EA		3883	ALC	FZICTR(1,@BR),FZIMI1(,@BR)	DECREMENT LOOP COUNTER						
	0892	D0	84 35		3884	BH	FZI009(,@BR)	DO NEXT DIGIT						
					3885	*								
					3886	*	AT THE CONCLUSION OF I@PREC ITERATIONS, THE I@PREC MOST SIGNIFICANT							
					3887	*	DIGITS OF THE PRODUCT OR THE MANTISSAS START EITHER AT I@1SE3+1(,@XR)							
					3888	*	OR AT I@1SE3+2(,@XR) AND END AT I@RSE3(,@XR) OR I@RSE3+1(,@XR) RE-							
					3889	*	SPECTIVELY, DEPENDING ON WHETHER I@1SE3+1(,@XR) IS NON-ZERO OR ZERO							
					3890	*	RESPECTIVELY. THE INDEX REG HAS BEEN DECREMENTED BY I@PREC AT THIS							
					3891	*	POINT. IF I@1SE3+1(1,@XR) DOES CONTAIN A ZERO, THE EXPONENT OF A IS							
					3892	*	DECREMENTED BY 1 TO COMPENSATE.							
					3893	*								
	0895	96	71 19 E8		3894	AZ	I@RSE3+FZIRD2(I@LUFV+1,@XR),FZIRD2(,@BR)	ROUND						
	0899	BD	F0 11		3895	CLI	I@1SE3+FZIMN1(,@XR),@DZERO	IS LEADING DIGIT ZERO ?						
	089C	D0	81 86		3896	BE	FZI060(,@BR)	BRANCH IF YES						
	089F	AC	06 0E 17		3897	MVC	I@RSE1+I@PREC(I@PREC,@XR),I@RSE3(,@XR)	MOVE RESULT->STACK						
	08A3	D0	87 8E		3898	B	FZI065(,@BR)	CONTINUE						
	08A6	AC	06 0E 18		3899	FZI060 MVC	I@RSE1+I@PREC(I@PREC,@XR),I@RSE3+FZIMN1(,@XR)	RESLT->STACK						
	08AA	9E	00 07 EA		3900	ALC	I@DEXP+I@PREC(1,@XR),FZIMI1(,@BR)	DECREMENT EXP BY 1						
	08AE	35	02 0D4E		3901	FZI065 L	IZSTAK,@XR	RESTORE INDEX REGISTER						
					3902	*								
					3903	*	CHECK FOR RESULT OF ZERO							
					3904	*								
	08B2	BD	F0 01		3905	CLI	I@1SE1+I@MANL(,@XR),@DZERO	IS LEADING DIGIT ZERO ?						
	08B5	D0	01 A2		3906	BNE	FZI070(,@BR)	BRANCH IF NO						
	08B8	BC	1E 00		3907	MVI	I@1SE1+I@DEXP(,@XR),B@NXLO	YES, SET EXP TO -98						
	08BB	A7	06 07 07		3908	SZ	I@RSE1(I@PREC,@XR),I@RSE1(I@PREC,@XR)	MANTISSA ALL ZEROES						
	08BF	D0	87 D4		3909	B	FZI888(,@BR)	EXIT						
					3910	*								
					3911	*	MAKE RESULT MINUS IF A AND B HAD DIFFERENT SIGNS							
					3912	*								
	08C2	75	04 DD		3913	FZI070 L	FZISUM(,@BR),@PSR	LOAD SIGN COMPARE BACK INTO PSR						
	08C5	D0	81 AB		3914	BE	FZI080(,@BR)	BRANCH IF SIGNS ARE EQUAL						
	08C8	BB	20 07		3915	SBF	I@RSE1(,@XR),X'20'	SET SIGN MINUS						
					3916	*								
					3917	*	CHECK FOR OVERFLOW OR UNDERFLOW							
					3918	*								
	08CB	7C	00 DC		3919	FZI080 MVI	FZISUM-FZIONE(,@BR),@ZERO	INSERT LEADING ZERO						
	08CE	6C	00 DD 00		3920	MVC	FZISUM(1,@BR),I@DEXP(,@XR)	FIRST EXP PLACED IN 2 BYTE ARE						
	08D2	5E	01 DD DF		3921	ALC	FZISUM(FZIEXP,@BR),FZIMUC-I@PREC(,@BR)	ADD SECOND EXPONEN						
	08D6	5D	01 DD EC		3922	CLC	FZISUM(FZIEXP,@BR),FZIUPL(,@BR)	IS RESULT TOO LARGE ?						
	08DA	D0	82 C1		3923	BL	FZI090(,@BR)	BRANCH IF NO						
	08DD	3C	ED 0CBC		3924	MVI	IZERRC,@@E791	YES, SET OVERFLOW FLAG						

## S/3 BASIC INTERPRETER - FLOATING MULTIPLY

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 06/09/20 PAGE 22

```

08E1 5D 01 DD F0      3925 FZI090 CLC   FZISUM(FZIEXP,@BR),FZILOL(@BR) IS RESULT TOO SMALL ?
08E5 D0 84 CC        3926          BH   FZI100(@BR)          BRANCH IF NO
08E8 3C EE 0CBC      3927          MVI   IZERRC,@E792          YES, SET UNDERFLOW FLAG
08EC 5F 01 DD EE      3928 FZI100 SLC   FZISUM(FZIEXP,@BR),FZINZR(@BR) SUBTRACT HEX '80'
08F0 9C 00 00 DD      3929          MVC   I@DEXP(1,@XR),FZISUM(@BR) MOVE NORMALIZED EXP TO RESULT
08F4 C2 01 0000      3930 FZI888 LA    *-*,@BR          RESTORE BASE REGISTER
08F8 C0 87 0000      3931 FZI890 B     *-*          RETURN TO CALLING ROUTINE
                    3932 *
                    3933 * CONSTANTS FOR FZIMPY EXECUTION
                    3934 *
0000 3935 FZIZRO EQU   0          BINARY LEADING ZERO
0001 3936 FZIONE EQU   1          LENGTH IN CLEARING ACCUMULATOR
0001 3937 FZIMN1 EQU   1          LENGTH FOR MOVING MANTISSA
0002 3938 FZIRD2 EQU   2          LENGTH FOR ROUNDING MANTISSA
0002 3939 FZIEXP EQU   2          LENGTH OF EXP SUM MOLDER
FFFF 3940 FZIMS1 EQU  -1          MINUS ONE
                    3941 *
                    3942 * WORK AREA FOR FZIMPY FOLLOWS
                    3943 *
08FC          08FD 3944 FZISUM DS    CL(FZIEXP)          FOR TEMPORARY USE - EXP SUM
08FE          08FE 3945 FZICTR DS    CL1          LOOP COUNTER
08FF          0906 3946 FZIMUC DS    XL(I@LUFV)          MULTIPLICAND
                    3947 *
                    3948 * THE FIRST ZERO OF THE ROUNDING INCREMENT IS USED TO RECOMPLEMENT
                    3949 *
0907 F0F5      0908 3950 FZIRDR DC    DL(FZIRD2)'05'          FOR ROUNDING RESULT
                    3951 *
                    3952 * CONSTANTS FOR FZIMPY FOLLOWS
                    3953 *
0909 FFFF      090A 3954 FZIMI1 DC    AL(FZIEXP)(FZIMS1)          MINUS ONE
090B 0164      090C 3955 FZIUPL DC    AL(FZIEXP)(B@NXZR+B@NXHI+1) UPPER LIMIT ON EXP SUM
090D 0080      090E 3956 FZINZR DC    AL(FZIEXP)(B@NXZR)          2 BYTE NORMALIZED ZERO
090F 009D      0910 3957 FZILOL DC    AL(FZIEXP)(B@NXZR+B@NXLO-1) LOWER LIMIT ON EXP SUM
                    3958 *
0919          3959          ORG    *+I@LUFL-I@LUFV          ADJUST FOR LONG PREC ROUTINE
                    3960 *
                    3961 * END OF FZIMPY CODING
                    3962 *

```

## S/3 BASIC DIVIDE RTN PROLOGUE: STANDARD &amp; LONG PREC.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20	PAGE 23
		3964		*****			
		3965	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
		3966	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
		3967	*				*
		3968		*****			*
		3969	*	STATUS			*
		3970	*	VERSION 1 MODIFICATION 0			*
		3971	*				*
		3972	*	FUNCTION			*
		3973	*	* FFIDVD DIVIDES THE SECOND INTERPRETER ELEMENT INTO THE FIRST			*
		3974	*	STACK ELEMENT.			*
		3975	*	* FFIDVD PERFORMS A STRAIGHT DIVISION PROCESS, EXCEPT THAT IN			*
		3976	*	ORDER TO SAVE TIME, IT HANDLES TWICE THE DIVISOR TO OBTAIN			*
		3977	*	APPROXIMATELY HALF AS MANY SUBTRACTIONS.			*
		3978	*	* FFIDVD RETURNS THE QUOTIENT IN THE FIRST STACK ELEMENT.			*
		3979	*				*
		3980	*	ENTRY POINTS			*
		3981	*	* THE ENTRY IS FFIDVD. THE FORMAT OF THE CALLING SEQUENCE IS AS			*
		3982	*	FOLLOWS:			*
		3983	*	B FFIDVD			*
		3984	*				*
		3985	*	INPUT			*
		3986	*	* THE INPUT IS TWO UNPACKED FLOATING POINT NUMBERS, IN THE FIRST			*
		3987	*	TWO INTERPRETER STACK ELEMENTS.			*
		3988	*	* THE ADDRESS RECALL REGISTER (ARR) IS STORED, AND CONTROL IS			*
		3989	*	RETURNED BY BRANCHING TO THE ADDRESS IN IT.			*
		3990	*				*
		3991	*	OUTPUT			*
		3992	*	* THE QUOTIENT IS LEFT IN THE FIRST STACK ELEMENT, IN UNPACKED			*
		3993	*	FLOATING POINT FORMAT.			*
		3994	*	* IN THE EVENT OF AN ERROR, THE APPROPRIATE CODE IS PLACED IN THE			*
		3995	*	INTERPRETER ONE-BYTE ERROR LOCATION IZERRC.			*
		3996	*				*
		3997	*	EXTERNAL REFERENCES			*
		3998	*	IZSTAK - LOCATION OF ADDRESS OF THE INTERPRETER STACK			*
		3999	*	INTERPRETER STACK - FIRST THREE ELEMENTS			*
		4000	*	IZERRC - ONE-BYTE INTERPRETER ERROR LOCATION			*
		4001	*				*
		4002	*	EXITS, NORMAL			*
		4003	*	* EXIT IS BY BRANCHING TO THE RETURN ADDRESS IN THE ADDRESS			*
		4004	*	RECALL REGISTER (ARR) STORED AT ENTRY.			*
		4005	*	* INDEX REGISTER 1 (@BR) IS RESTORED BEFORE RETURNING.			*
		4006	*	* THE QUOTIENT IS PLACED IN THE FIRST STACK ELEMENT AS REFERENCED			*
		4007	*	BY THE ADDRESS IN IZSTAK.			*
		4008	*				*
		4009	*	EXITS, ERROR			*
		4010	*	* AN ERROR CODE IS PLACED IN THE INTERPRETER ERROR AREA, IZERRC.			*
		4011	*	* EXIT IS BY BRANCHING TO THE RETURN ADDRESS OF THE STORED ARR.			*
		4012	*				*
		4013	*	TABLES/WORK AREA			*
		4014	*	THE CONSTANTS AND WORK AREA RESIDE AT THE END OF THE EXECUTABLE			*
		4015	*	CODE, AND ARE REFERENCED BY A DISPLACEMENT RELATIVE TO THE VALUE			*
		4016	*	IN @BR.			*
		4017	*				*
		4018	*	ATTRIBUTES			*
		4019	*	REUSABLE, RELOCATABLE.			*



S/3 BASIC DIVIDE RTN PROLOGUE: STANDARD & LONG PREC.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20	PAGE 24
		4020	*				*
		4021	*	*CHARACTER CODE DEPENDENCY			*
		4022	*	THE OPERATION OF THIS MODULE DEPENDS UPON A ZONED DECIMAL DIGIT			*
		4023	*	BEING REPRESENTED WITH THE ZONE (TIRST FOUR BITS) BEING AN 'F'			*
		4024	*	FOR POSITIVE, AND A 'D' FOR NEGATIVE. THE DECIMAL NUMBERS MUST			*
		4025	*	BE CODED SO THAT THE LOW ORDER FOUR BITS, WHEN CONSIDERED AS A			*
		4026	*	BINARY INTEGER, IDENTITY TWE VALLE OF THE DIGIT.			*
		4027	*	THESE PROPERTIES ARE USED AT FFI003, AND FOLLOWING FFI030.			*
		4028	*				*
		4029	*	*NOTES			*
		4030	*	ERROR PROCEDURES			*
		4031	*	THE ERROR CODE IS SET, AND CONTROL RETURNED TO THE CALLING			*
		4032	*	PROGRAM.			*
		4033	*				*
		4034	*	REGISTER USAGE			*
		4035	*	INDEX REGISTER 1 (@BR) IS SAVED AND RESTORED. THIS REGISTER			*
		4036	*	IS USED AS THE BASE REGISTER DURING EXECUTION.			*
		4037	*	INDEX REGISTER 2 (@XR) IS LOADED TO CONTAIN THE ADDRESS OF			*
		4038	*	THE FIRST BYTE OF THE INTERPRETER STACK, TO REFERENCE IT,			*
		4039	*	@XR IS NOT SAVED OR RESTORED.			*
		4040	*				*
		4041	*	SAVED/RESTORED AREAS			*
		4042	*	NONE.			*
		4043	*				*
		4044	*	MODIFICATION CONSIDERATIONS			*
		4045	*	NONE.			*
		4046	*				*
		4047	*	REQUIRED MODULES			*
		4048	*	@SYSEQ - COMMON SYSTEM EQUATES			*
		4049	*	@ERMEQ - ERROR MESSAGE EQUATES			*
		4050	*	\$I@SEQ - STANDARD PRECISION EXECUTION EQUATES			*
		4051	*	\$B@EQU - COMPILER SYSTEM EQUATES			*
		4052	*	IZCOMN - CORE RESIDENT COMMON LOCATION EQUATES			*
		4053	*				*
		4054	*	OTHER			*
		4055	*	NONE.			*
		4056	*	*****			*

S/3 BASIC DIVIDE RTN PROLOGUE: STANDARD & LONG PREC.

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 25
			4058		*****	
			4059		* FFIDVD - S/3.7 DIVIDE FLOATING POINT ROUTINE	
			4060		*****	
			0937 4061		USING FFI002,@BR SET ADDRESSABILITY	
			4062	*		
			4063	*	EXECUTION ENTRY TO FFIDVD FOLLOWS	
			4064	*		
0919	34	01 09F4	4065	FFIDVD ST	FFI888+@OP1,@BR SAVE BASE REG	
091D	C2	01 0937	4066	LA	FFI002,@BR LOAD BASE REGISTER	
0921	74	08 C1	4067	ST	FFI890+@OP1(,@BR),@ARR SAVE RETURN ADDRESS	
0924	35	02 0D4E	4068	L	IZSTAK,@XR LOAD STACK POINTER	
			4069	*		
			4070	*	CHECK FOR ZERO DIVISOR	
			4071	*		
0928	BD	F0 09	4072	CLI	I@1SE2+I@MANL(,@XR),@DZERO IS DIVISOR ZERO ?	
092B	7C	EC AF	4073	MVI	FFI350+@Q(,@BR),@E790 SET DIVIDE BY ZERO ERROR FLAG	
092E	D0	81 AE	4074	BE	FFI350(,@BR) IF DIVISOR = 0, RETURN ERROR	
			4075	*		
			4076	*	IF DIVIDEND IS ZERO, LEAVE IT AS RESULT AND EXIT	
			4077	*		
0931	BD	F0 01	4078	FFI001 CLI	I@1SE1+I@MANL(,@XR),@DZERO IS DIVIDEND ZERO ?	
0934	D0	81 BA	4079	BE	FFI888(,@BR) YES, LEAVE RESULT ZERO AND EXIT	
0937	6C	07 D7 0F	4080	FFI002 MVC	FFIDIV(I@LUFV,@BR),I@RSE2(,@XR) SAVE DIVISOR	
093B	6C	00 CE 00	4081	MVC	FFIXPO(1,@BR),I@1SE1+I@DEXP(,@XR) SAVE EXP OF DIVIDEND	
			4082	*		
			4083	*	INSERT DECIMAL ZERO IN FRONT OF DIVIDEND SO IT CAN BE USED AS A	
			4084	*	ZONED DECIMAL NUMBER DURING THE ALGORITHM	
			4085	*		
093F	BC	F0 00	4086	MVI	I@1SE1(,@XR),@DZERO INSERT LEADING ZERO	
			4087	*		
			4088	*	TO COMPARE SIGNS, MAKE THE NUMERIC PARTS OF THE BYTES CONTAINING	
			4089	*	THE SIGN ZONE THE SAME IN BOTH DIVISOR AND DIVIDEND SO THAT A	
			4090	*	COMPARE LOGICAL INSTRUCTION CAN BE USED. THE NUMERIC OF THE DIVISOR	
			4091	*	IS DESTROYED, BUT THE DIVISOR WAS SAVED EARLIER.	
			4092	*		
0942	A8	03 0F 07	4093	MNN	I@RSE2(,@XR),I@RSE1(,@XR) SET NUMERIC OF B TO THAT OF A	
0946	AD	00 0F 07	4094	CLC	I@RSE2(1,@XR),I@RSE1(,@XR) COMPARE SIGNS	
094A	74	04 CC	4095	ST	FFIPSR(,@BR),@PSR SAVE CONDITION REG	
094D	7A	F0 D7	4096	FFI003 SBN	FFIDIV(,@BR),B@ZPOS FORCE DIVISOR POSITIVE	
0950	BA	F0 07	4097	SBN	I@RSE1(,@XR),B@ZPOS FORCE DIVIDEND POSITIVE	
			4098	*		
			4099	*	LOAD ACCUMULATOR WITH DECIMAL 2'S. THE FIRST PART OF AN ITERATION	
			4100	*	INCREMENTS THE ASSOCIATED DIGIT OF THE RESULT (INITIALLY 2) BY 2	
			4101	*	WHENEVER A SUBTRACTION OR TWICE THE DIVISOR FROM THE DIVIDEND YIELDS	
			4102	*	A POSITIVE RESULT.	
			4103	*		
0953	BC	F2 18	4104	MVI	I@RSE3+1(,@XR),B@DEC2 GET DECIMAL 2	
0956	AC	07 17 18	4105	MVC	I@RSE3(I@LUFV,@XR),I@RSE3+1(,@XR) PROPAGATE 2'S	
095A	A4	70 0F 00	4106	ZAZ	I@RSE2(I@LUFV,@XR),I@1SE1(1,@XR) SET ACCUMULATOR TO ZERO	
			4107	*		
			4108	*	DOUBLE THE DIVISOR FOR USE AS A SUBTRACTED DURING THE FIRST PART	
			4109	*	OF AN ITERATION.	
			4110	*		
095E	54	16 DF D7	4111	ZAZ	FFI2DV(I@LUFV,@BR),FFIDIV(I@PREC,@BR) MANTISSA W/ LEAD 0	
0962	56	07 DF DF	4112	AZ	FFI2DV(I@LUFV,@BR),FFI2DV(I@LUFV,@BR) DOUBLE THE DIVISOR	
			4113	*		

## S/3 BASIC DIVIDE RTN PROLOGUE: STANDARD &amp; LONG PREC.

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 06/09/20 PAGE 26

```

4114 * IF THE MANTISSA OF THE DIVIDEND IS LESS THEN THAT OF THE DIVISOR, A
4115 * LOOP OF I@LUFV ITERATIONS IS USED TO INSURE ROUNDING THE ANSWER TO
4116 * I$PREC DIGITS. OTHERWISE, I@PREC ITERATIONS ARE USED.
4117 *
0966 9D 06 07 D7 4118 CLC I@RSE1(I@PREC,@XR),FFIDIV(,@BR) IS DIVIDEND < DIVISOR ?
096A D0 02 3F 4119 BNL FFI004(,@BR) NO, BRANCH TO SET LOOP = I@PREC
096D 7C 09 CD 4120 MVI FFICNT(,@BR),I@LUFV+1 SET COUNTER TO I@LUFV ITERATIONS
0970 7C 87 6D 4121 MVI FFI015+@Q(,@BR),@UCB SET BRANCH TO REFERENCE ANSWER
0973 D0 87 4C 4122 B FFI006(,@BR) BRANCH TO ITERATION
0976 7C 08 CD 4123 FFI004 MVI FFICNT(,@BR),I@LUFV SET COUNTER TO I@PREC ITERATIONS
0979 7C 80 6D 4124 MVI FFI015+@Q(,@BR),@NOP SET BRANCH TO A NOP FOR ANSWER
097C D0 87 4C 4125 B FFI006(,@BR) BRANCH TO ITERATION
4126 *
4127 * START FIRST PART OF ITERATION
4128 *
097F 9E 00 10 C2 4129 FFI005 ALC I@1SE3(1,@XR),FFIIN2(,@BR) ADD 2 TO ACCUMULATOR
0983 97 07 07 DF 4130 FFI006 SZ I@RSE1(I@LUFV,@XR),FFI2DV(I@LUFV,@BR) SUBTRACT 2*DIVISOR
0987 D0 02 48 4131 BNM FFI005(,@BR) STILL POSITIVE, REPEAT
4132 *
4133 * START SECOND PART OF ITERATION
4134 *
098A 9F 00 10 C3 4135 FFI010 SLC I@1SE3(1,@XR),FFIIN1(,@BR) SUB 1 FROM ACCUMULATOR
098E 96 16 07 D7 4136 AZ I@RSE1(I@LUFV,@XR),FFIDIV(I@PREC,@BR) ADD THE DIVISOR
0992 D0 82 53 4137 BM FFI010(,@BR) STILL NEGATIVE
4138 *
4139 * END OF ITERATION
4140 *
0995 E2 02 01 4141 FFI011 LA FFIPTR(,@XR),@XR INCREMENT POINTER
0998 5F 00 CD C3 4142 SLC FFICNT(1,@BR),FFIIN1(,@BR) DECREMENT LOOP COUNT
099C D0 84 4C 4143 BH FFI006(,@BR) COMPUTE NEXT DIGIT
099F 35 02 0D4E 4144 L IZSTAK,@XR RESTORE INDEX REGISTER
4145 *
4146 * END OF ALGORITHM. NORMALIZE RESULT AND INSERT CORRECT SIGN.
4147 *
4148 * SMALLEST POSSIBLE RESULT IS 0.10000000/0.99999999 = 0.10000000, AND
4149 * LARGEST POSSIBLE RESULT IS 0.99999999/0.10000000 = 9.99999999, SO
4150 * THAT IF LEADING DIGIT IS NON-ZERO WE HAVE A RESULT OF THE SECOND
4151 * TYPE. IN THIS CASE, THE RESULT IS I@PREC+1 DIGITS LONG SO THAT
4152 * ONLY THE FIRST I@PREC DIGITS ARE USED AND THE EXPONENT IS INCREMENTED
4153 * BY 1 TO NORMALIZE.
4154 *
09A3 D0 00 7E 4155 FFI015 BC FFI020(,@BR),*-* BRANCH IF DIVDND MNTSSA < DIVSR
09A6 96 70 17 C6 4156 AZ I@RSE3(I@LUFV,@XR),FFIRND(,@BR) ROUND THE ANSWER
09AA AC 06 07 16 4157 MVC I@RSE1(I@PREC,@XR),I@RSE3-1(,@XR) QUOTIENT INTO STACK
09AE 5E 00 CE C3 4158 ALC FFI015+@Q(,@BR),@UCB ADD 1 TO EXPONENT OF A
09B2 D0 87 86 4159 B FFI030(,@BR) CALCULATE EXPONENT OF QUOTIENT
09B5 96 70 18 C6 4160 FFI020 AZ I@RSE3+1(I@LUFV,@XR),FFIRND(1,@BR) ROUND THE ANSWER
09B9 AC 06 07 17 4161 MVC I@RSE1(I@PREC,@XR),I@RSE3(,@XR) MOVE QUOTIENT INTO STACK
09BD 75 04 CC 4162 FFI030 L FFIPSR(,@BR),@PSR LOAD SIGN COMPARE BACK INTO PSR
09C0 D0 81 8F 4163 BE FFI200(,@BR) BRANCH IF SIGNS EQUAL
09C3 BB 20 07 4164 SBF I@RSE1(,@XR),X'20' SET SIGN MINUS
4165 *
4166 * COMPUTE EXPONENT OF RESULT FROM EXPONENTS OF A AND B, AND CHECK
4167 * FOR OVERFLOW OR UNDERFLOW.
4168 *
09C6 7C 01 CD 4169 FFI200 MVI FFI015-FFIAPB(,@BR),X'01' ADD HEX 100 TO EXPONENT OF A

```

## S/3 BASIC DIVIDE RTN PROLOGUE: STANDARD &amp; LONG PREC.

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20 PAGE 27
09C9	5F 01 CE D0		4170	SLC	FFIXPO(FFIEXP,@BR),FFIDIV-I@PREC(,@BR)	SUBTRACT EXP OF B	
09CD	5D 01 CE C5		4171	CLC	FFIXPO(FFIEXP,@BR),FFIXHI(,@BR)	IS RESULT TOO LARGE ?	
09D1	7C ED AF		4172	MVI	FFI350+@Q(,@BR),@@E791	OVERFLOW FLAG	
09D4	D0 02 AE		4173	BNL	FFI350(,@BR)	RETURN TO CALLING PROGRAM	
09D7	5D 01 CE C8		4174	FFI300 CLC	FFIXPO(FFIEXP,@BR),FFIXLO(,@BR)	IS RESULT TOO SMALL ?	
09DB	3C 00 0CBC		4175	MVI	IZERRC,I@NERR	CLEAR OUT ANY FALSE ERROR FLAGS	
09DF	D0 84 B2		4176	BH	FFI400(,@BR)	NO, BRANCH	
09E2	7C EE AF		4177	MVI	FFI350+@Q(,@BR),@@E792	SET UNDERFLOW FLAG	
09E5	3C 00 0CBC		4178	FFI350 MVI	IZERRC,*-*	SET ERROR FLAG	
09E9	5F 01 CE CA		4179	FFI400 SLC	FFIXPO(FFIEXP,@BR),FFIZRO(,@BR)	SUBTRACT NORMALIZED 0	
09ED	9C 00 00 CE		4180	MVC	I@1SE1+I@DEXP(1,@XR),FFIXPO(,@BR)	INSERT NORMALIZED EXP	
09F1	C2 01 0000		4181	FFI888 LA	*-*,@BR	RESTORE BASE REG	
09F5	C0 87 0000		4182	FFI890 B	*-*	RETURN TO CALLING PROGRAM	
			4183	*			
			4184	*	CONSTANTS FOR FFIDVD FOLLOW		
			4185	*			
		0001	4186	FFIAC1 EQU	1	CONSTANT IN PROPAGATING-ACCUMU.	
		0001	4187	FFIPTR EQU	1	INCREMENT FOR NUMBER PROCESSING	
		0001	4188	FFIONE EQU	1	CONSTANT OF 1	
		0001	4189	FFIAPB EQU	1	TO ADD A PLUS B	
		0002	4190	FFIEXP EQU	2	LENGTH OF EXP SUM AND WORK AREA	
			4191	*			
			4192	*	CONSTANTS FOR FFIDVD FOLLOW		
			4193	*			
09F9	02	09F9	4194	FFIIN2 DC	XL1'02'	INCREMENT OF 2	
09FA	01	09FA	4195	FFIIN1 DC	AL1(@B1)	INCREMENT OF 1	
09FB	0164	09FC	4196	FFIXHI DC	AL(FFIEXP)(B@NXZR+B@NXHI+1)	UPPER LIMIT FOR 2 BYTE EXP	
09FD	F5	09FD	4197	FFIRND DC	DL1'5'	DECIMAL 5 FOR ROUNDING	
09FE	009D	09FF	4198	FFIXLO DC	AL(FFIEXP)(B@NXZR+B@NXLO-1)	LOWER LIMIT FOR 2 BYTE EXP	
0A00	0080	0A01	4199	FFIZRO DC	AL(FFIEXP)(B@NXZR)	NORMALIZED ZERO	
			4200	*			
			4201	*	WORK AREA FOR FFIDVD FOLLOWS		
			4202	*			
0A02		0A03	4203	FFIPSR DS	CL(@REGL)	CONDITION REGISTER (PSR)	
0A04		0A04	4204	FFICNT DS	CL1	LOOP COUNTER	
0A05		0A05	4205	FFIXPO DS	CL1	FOR EXPONENT MANIPULATION	
0A06	00	0A06	4206	DC	XL1'0'	LEADING ZERO FOR EXPONENT	
0A07		0A0E	4207	FFIDIV DS	CL(I@LUFV)	DIVISOR	
0A0F		0A16	4208	FFI2DV DS	CL(I@LUFV)	TWICE DIVISOR	
			4209	*			
0A27			4210	ORG	*+2*I@LUFL-2*I@LUFV	ADJUST FOR LONG PREC ROUTINE	
			4211	*			
			4212	*	END OF FFIDVD CODING		
			4213	*			

## S/3 BASIC INTERPRETER FLOATING POINT UNPACKER

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 28
		4215		*****	
		4216	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		4217	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
		4218	*		*
		4219		*****	
		4220		*STATUS	*
		4221	*	VERSION 1 MODIFICATION 0	*
		4222	*		*
		4223		*FUNCTION	*
		4224	*	* CPUFLT CONVERTS STANDARD OR LONG PRECISION PACKED FLOATING	*
		4225	*	POINT DECIMAL VALUES TO UNPACKED FLOATING POINT DECIMAL VALUES	*
		4226	*	SUITABLE FOR ARITHMETIC OPERATIONS.	*
		4227	*	* A PACKED DECIMAL FLOATING POINT VALUE REFERENCED BY REGISTER	*
		4228	*	@XR IS CONVERTED TO AN UNPACKED (ZONED) DECIMAL FLOATING POINT	*
		4229	*	VALUE. THE CONVERTED, VALUE IS LEFT IN THE LOCATION ORIGINALLY	*
		4230	*	OCCUPIED BY THE PACKED VALUE. REGISTER @XR IS NOT MODIFIED	*
		4231	*	DURING EXECUTION.	*
		4232	*		*
		4233		*ENTRY POINTS	*
		4234	*	* THIS ROUTINE HAS A SINGLE ENTRY POINT - CPUFLT - WHOSE FUNCTION	*
		4235	*	IS DEFINED ABOVE. CALLING SEQUENCE IS	*
		4236	*	B CPUFLT	*
		4237	*	SUBJECT TO INPUT CONDITIONS DESCRIBED BELOW.	*
		4238	*	* ENTRY POINT CPUFLT MAY ALSO BE SPECIFIED AS ISCPUF WHEN CALLED	*
		4239	*	FROM ONE OF THE SUBROUTINES RESIDENT IN VIRTUAL MEMORY.	*
		4240	*		*
		4241		*INPUT	*
		4242	*	REGISTER @XR - FOR THE PACKED FLOATING POINT VALUE POINTER.	*
		4243	*	THIS CONTAINS THE CORE ADDRESS OF THE LEFTMOST BYTE OF THE	*
		4244	*	FLOATING POINT VALUE TO BE UNPACKED.	*
		4245	*		*
		4246		*OUTPUT	*
		4247	*	UNPACKED FLOATING POINT VALUE - LOCATED WITH LEFTMOST BYTE	*
		4248	*	REFERENCED BY REGISTER @XR, REPLACING THE ORIGINAL PACKED	*
		4249	*	FLOATING POINT VALUE.	*
		4250	*		*
		4251		*EXTERNAL REFERENCES	*
		4252	*	N/A	*
		4253	*		*
		4254		*EXITS, NORMAL	*
		4255	*	CONTROL IS ALWAYS RETURNED TO THE FIRST INSTRUCTION FOLLOWING THE	*
		4256	*	CPUFLT CALLING SEQUENCE.	*
		4257	*		*
		4258		*EXITS, ERROR	*
		4259	*	N/A	*
		4260	*		*
		4261		*TABLES/WORK AREAS	*
		4262	*	CONVERSION REQUIRES A TEMPORARY WORK AREA ADJACENT TO THE ORIGI-	*
		4263	*	NAL PACKED FLOATING POINT VALUE. REGION DISPLACEMENTS (RELATIVE	*
		4264	*	TO REGISTER @XR) FOR EACH STEP IN THE CONVERSION ARE -	*
		4265	*	* ORIGINAL PACKED VALUE STD - 0 TO 4, LONG - 0 TO 8	*
		4266	*	* TEMPORARY WORK AREA STD - 8 TO 12, LONG - 16 TO 24	*
		4267	*	* FINAL UNPACKED VALUE STD - 0 TO 7, LONG - 0 TO 15	*
		4268	*	THE ORIGINAL CONTENTS OF THE TEMPORARY WORK AREA ARE NOT SAVED.	*
		4269	*		*
		4270		*ATTRIBUTES	*



## S/3 BASIC INTERPRETER FLOATING POINT UNPACKER

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 29
		4271	*	* REUSABLE	*
		4272	*	* RELOCATABLE	*
		4273	*		*
		4274	*	*CHARACTER CODE DEPENDENCY	*
		4275	*	THE OPERATION OF THIS MODULE DEPENDS UPON THE FOLLOWING PROPER-	*
		4276	*	TIES OF THE INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET	*
		4277	*	* MOST CODING HAS BEEN ARRANGED SO THAT REDEFINITION OF CHAR-	*
		4278	*	ACTER CONSTANTS, BY REASSEMBLY, WILL RESULT IN A CORRECT	*
		4279	*	MODULE FOR THE NEW DEFINITION.	*
		4280	*	* NUMERIC CHARACTERS 0 THROUGH 9 ARE PRESUMED TO BE CODED SUCH	*
		4281	*	THAT THE HIGH ORDER FOUR BITS CONTAIN A SIGN ZONE WITH X'F'	*
		4282	*	DEFINING A POSITIVE DIGIT AND X'D' DEFINING A NEGATIVE DIGIT.	*
		4283	*	* DECIMAL NUMBERS MUST ALSO BE CODED SO THAT THE LOW ORDER	*
		4284	*	FOUR BITS, WHEN CONSIDERED AS A BINARY INTEGER, IDENTIFY THE	*
		4285	*	VALUE OF THE DIGIT.	*
		4286	*	THE SPECIFIC INSTRUCTIONS (INSTRUCTION SEQUENCES) WHICH REQUIRE	*
		4287	*	MODIFICATION IF THESE PROPERTIES OF THE CHARACTER SET ARE CHANGED	*
		4288	*	BY BE IDENTIFIED BY -	*
		4289	*	* THE SINGLE INSTRUCTION JUST PRIOR TO LABEL CPU050.	*
		4290	*	* THE 7 INSTRUCTIONS BEGINNING AT LABEL CPU050.	*
		4291	*	* THE 8 INSTRUCTIONS BEGINNING AT LABEL CPU070.	*
		4292	*		*
		4293	*	*NOTES	*
		4294	*	ERROR PROCEDURES	*
		4295	*	N/A	*
		4296	*		*
		4297	*	REGISTER USAGE	*
		4298	*	* REGISTER @BR IS NOT USED.	*
		4299	*	* REGISTER @XR IS USED AS AN INPUT PARAMETER, AND RETAINS THIS	*
		4300	*	SAME VALUE AT CPUFLT EXIT.	*
		4301	*		*
		4302	*	SAVED/RESTORED AREAS	*
		4303	*	N/A	*
		4304	*		*
		4305	*	MODIFICATION CONSIDERATIONS	*
		4306	*	CPUFLT UTILIZES SEQUENTIAL INSTRUCTIONS RATHER THAN A LOOPING	*
		4307	*	TECHNIQUE FOR MANTISSA UNPACKING. THE ORDER AND SEQUENCE OF	*
		4308	*	THESE INSTRUCTIONS THEREFORE DEPENDS ON THE NUMBER OF SIGNIFI-	*
		4309	*	CANT DIGITS IN THE MANTISSA.	*
		4310	*		*
		4311	*	REQUIRED MODULES	*
		4312	*	* @SYSEQ - COMMON SYSTEM EQUATES.	*
		4313	*	* @B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.	*
		4314	*	* \$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD. PREC. ONLY).	*
		4315	*	* \$I@LEQ - INTERPRETER PARAMETER EQUATES (FOR LONG PREC. ONLY).	*
		4316	*		*
		4317	*	OTHER	*
		4318	*	N/A	*
		4319	*	*****	*

## S/3 BASIC INTERPRETER FLOATING POINT UNPACKER

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 30
			4321		*****	
			4322		* FLOATING POINT VALUE UNPACKING ROUTINE ENTRY POINT	
			4323		*****	
			4324		*	
			4325		* ENTER CPUFLT - SAVE THE RETURN ADDRESS	
			4326		*	
0A27	34 08 0A83		0A27	4327	CPUFLT EQU * CPUFLT ENTRY POINT	
				4328	ST CPU080+@OP1,@ARR SET RETURN BRANCH ADDRESS	
				4329	*	
				4330	* REGISTER @XR CONTAINS THE ADDRESS OF THE FLOATING POINT VALUE TO BE	
				4331	* UNPACKED - MOVE THE PACKED VALUE TO A SAVE AREA IN THE STACK	
				4332	*	
0A2B	AC 04 0C 04			4333	CPU010 MVC CPUSAV+I@LPFV-1(,@XR),I@LPFV-1(I@LPFV,@XR) SAVE PKD VALUE	
				4334	*	
				4335	* ESTABLISH EXPONENT FOR THE UNPACKED FLOATING POINT VALUE	
				4336	*	
0A2F	AC 00 00 0C			4337	CPU020 MVC I@UEXP(,@XR),CPUSAV+I@PEXP(1,@XR) MOVE EXP INTO UNPKD VAL	
				4338	*	
				4339	* INITIALIZE THE UNPACKED VALUE MANTISSA TO DECIMAL ZEROS	
				4340	*	
0A33	84 60 07 0A84			4341	CPU030 ZAZ I@UMNR(I@PREC,@XR),CPUDC0(1) FILL MANTISSA WITH DEC ZEROS	
				4342	*	
				4343	* ESTABLISH THE SIGN ZONE FOR THE UNPACKED VALUE	
				4344	*	
0A38	B9 10 08			4345	CPU040 TBF CPUSAV+I@STAT(,@XR),B@SIGN IF PACKED STATUS IS POSITIVE	
0A3B	F2 10 03			4346	JT CPU050 * GO UNPACK THE VALUE MANTISSA	
0A3E	BC D0 07			4347	MVI I@SIGN(,@XR),B@ZNEG * ELSE SET NEGATIVE SIGN ZONE	
				4348	*	
				4349	* PERFORM MANTISSA UNPACKING FOR STANDARD PRECISION DIGITS	
				4350	*	
0A41	A8 03 01 08			4351	CPU050 MNN CPUU01(,@XR),CPUP01(,@XR) UNPACK HIGH ORDER MANTISSA DIG	
0A45	A8 02 02 09			4352	MNZ CPUU02(,@XR),CPUP02(,@XR) UNPACK 2ND HIGH MANTISSA DIGIT	
0A49	A8 03 03 09			4353	MNN CPUU03(,@XR),CPUP03(,@XR) UNPACK 3RD HIGH MANTISSA DIGIT	
0A4D	A8 02 04 0A			4354	MNZ CPUU04(,@XR),CPUP04(,@XR) UNPACK 4TH HIGH MANTISSA DIGIT	
0A51	A8 03 05 0A			4355	MNN CPUU05(,@XR),CPUP05(,@XR) UNPACK 5TH HIGH MANTISSA DIGIT	
0A55	A8 02 06 0B			4356	MNZ CPUU06(,@XR),CPUP06(,@XR) UNPACK 6TH HIGH MANTISSA DIGIT	
0A59	A8 03 07 0B			4357	MNN CPUU07(,@XR),CPUP07(,@XR) UNPACK 7TH HIGH MANTISSA DIGIT	
				4358	*	
				4359	* TEST FOR EXECUTION PRECISION MODE	
				4360	*	
0A5D	F2 87 20			4361	CPU060 JC CPU080,I@PRSW BRANCH IF STANDARD PRECISION	
				4362	*	
				4363	* PERFORM MANTISSA UNPACKING FOR LONG PRECISION DIGITS	
				4364	*	
0A60	A8 02 08 0C			4365	CPU070 MNZ CPUU08(,@XR),CPUP08(,@XR) UNPACK 8TH HIGH MANTISSA DIGIT	
0A64	A8 03 09 0C			4366	MNN CPUU09(,@XR),CPUP09(,@XR) UNPACK 9TH HIGH MANTISSA DIGIT	
0A68	A8 02 0A 0D			4367	MNZ CPUU10(,@XR),CPUP10(,@XR) UNPACK 10TH HIGH MANTISSA DIGIT	
0A6C	A8 03 0B 0D			4368	MNN CPUU11(,@XR),CPUP11(,@XR) UNPACK 11TH HIGH MANTISSA DIGIT	
0A70	A8 02 0C 0E			4369	MNZ CPUU12(,@XR),CPUP12(,@XR) UNPACK 12TH HIGH MANTISSA DIGIT	
0A74	A8 03 0D 0E			4370	MNN CPUU13(,@XR),CPUP13(,@XR) UNPACK 13TH HIGH MANTISSA DIGIT	
0A78	A8 02 0E 0F			4371	MNZ CPUU14(,@XR),CPUP14(,@XR) UNPACK 14TH HIGH MANTISSA DIGIT	
0A7C	A8 03 0F 0F			4372	MNN CPUU15(,@XR),CPUP15(,@XR) UNPACK 15TH HIGH MANTISSA DIGIT	
				4373	*	
				4374	* RETURN CONTROL TO THE CALLING PROGRAM	
				4375	*	
0A80	C0 87 0000			4376	CPU080 B *- * RETURN TO CALLING PROGRAM	

## S/3 BASIC INTERPRETER FLOATING POINT UNPACKER

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 31
			4378	*****		
			4379	* FLOATING POINT UNPACKING ROUTINE CONSTANTS		
			4380	*****		
			4381	*		
0A84	F0		0A84	4382	CPUDC0 DC DL1'0' DECIMAL ZERO	
			4383	*		
			4384	*****		
			4385	* FLOATING POINT UNPACKING ROUTINE EQUATES		
			4386	*****		
			4387	*		
			0008	4388	CPUSAV EQU I@LUFV DISP FOR PACKED SAVE AREA	
			4389	*		
			0008	4390	CPUP01 EQU CPUSAV+0 DISP FOR PACKED 1ST DIGIT	
			0009	4391	CPUP02 EQU CPUSAV+1 DISP FOR PACKED 2ND DIGIT	
			0009	4392	CPUP03 EQU CPUSAV+1 DISP FOR PACKED 3RD DIGIT	
			000A	4393	CPUP04 EQU CPUSAV+2 DISP FOR PACKED 4TH DIGIT	
			000A	4394	CPUP05 EQU CPUSAV+2 DISP FOR PACKED 9TH DIGIT	
			000B	4395	CPUP06 EQU CPUSAV+3 DISP FOR PACKED 6TH DIGIT	
			000B	4396	CPUP07 EQU CPUSAV+3 DISP FOR PACKED 7TH DIGIT	
			4397	*		
			000C	4398	CPUP08 EQU CPUSAV+4 DISP FOR PACKED 8TH DIGIT	
			000C	4399	CPUP09 EQU CPUSAV+4 DISP FOR PACKED 9TH DIGIT	
			000D	4400	CPUP10 EQU CPUSAV+5 DISP FOR PACKED 10TH DIGIT	
			000D	4401	CPUP11 EQU CPUSAV+5 DISP FOR PACKED 11TH DIGIT	
			000E	4402	CPUP12 EQU CPUSAV+6 DISP FOR PACKED 12TH DIGIT	
			000E	4403	CPUP13 EQU CPUSAV+6 DISP FOR PACKED 13TH DIGIT	
			000F	4404	CPUP14 EQU CPUSAV+7 DISP FOR PACKED 14TH DIGIT	
			000F	4405	CPUP15 EQU CPUSAV+7 DISP FOR PACKED 15TH DIGIT	
			4406	*		
			0001	4407	CPUU01 EQU I@UMN1+0 DISP FOR UNPACKED 1ST DIGIT	
			0002	4408	CPUU02 EQU I@UMN1+1 DISP FOR UNPACKED 2ND DIGIT	
			0003	4409	CPUU03 EQU I@UMN1+2 DISP FOR UNPACKED 3RD DIGIT	
			0004	4410	CPUU04 EQU I@UMN1+3 DISP FOR UNPACKED 4TH DIGIT	
			0005	4411	CPUU05 EQU I@UMN1+4 DISP FOR UNPACKED 5TH DIGIT	
			0006	4412	CPUU06 EQU I@UMN1+5 DISP FOR UNPACKED 6TH DIGIT	
			0007	4413	CPUU07 EQU I@UMN1+6 DISP FOR UNPACKED 7TH DIGIT	
			4414	*		
			0008	4415	CPUU08 EQU I@UMN1+7 DISP FOR UNPACKED 8TH DIGIT	
			0009	4416	CPUU09 EQU I@UMN1+8 DISP FOR UNPACKED 9TH DIGIT	
			000A	4417	CPUU10 EQU I@UMN1+9 DISP FOR UNPACKED 10TH DIGIT	
			000B	4418	CPUU11 EQU I@UMN1+10 DISP FOR UNPACKED 11TH DIGIT	
			000C	4419	CPUU12 EQU I@UMN1+11 DISP FOR UNPACKED 12TH DIGIT	
			000D	4420	CPUU13 EQU I@UMN1+12 DISP FOR UNPACKED 13TH DIGIT	
			000E	4421	CPUU14 EQU I@UMN1+13 DISP FOR UNPACKED 14TH DIGIT	
			000F	4422	CPUU15 EQU I@UMN1+14 DISP FOR UNPACKED 15TH DIGIT	
			4423	*		
			4424	*****		
			4425	*		
			4426	* END OF FLOATING POINT UNPACKING ROUTINE CODING		
			4427	*		



## S/3 BASIC INTERPRETER FLOATING POINT PACKER

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20	PAGE 32
		4429		*****			
		4430	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
		4431	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
		4432	*				*
		4433		*****			*
		4434	*	*STATUS			*
		4435	*	VERSION 1 MODIFICATION 0			*
		4436	*				*
		4437	*	*FUNCTION			*
		4438	*	* CUPFLT CONVERTS STANDARD OR LONG PRECISION UNPACKED FLOATING			*
		4439	*	POINT DECIMAL VALUES TO PACKED FLOATING POINT DECIMAL VALUES			*
		4440	*	SUITABLE FOR STORAGE IN VIRTUAL STORAGE.			*
		4441	*	* A UNPACKED (ZONED) DECIMAL FLOATING POINT VALUE REFERENCED BY			*
		4442	*	REGISTER @XR IS CONVERTED TO AN PACKED DECIMAL FLOATING POINT			*
		4443	*	VALUE. THE CONVERTED, VALUE IS LEFT IN THE LOCATION ORIGINALLY			*
		4444	*	OCCUPIED BY THE UNPACKED VALUE. REGISTER @XR IS NOT MODIFIED			*
		4445	*	DURING EXECUTION.			*
		4446	*				*
		4447	*	*ENTRY POINTS			*
		4448	*	* THIS ROUTINE HAS A SINGLE ENTRY POINT - CUPFLT - WHOSE FUNCTION			*
		4449	*	IS DEFINED ABOVE. CALLING SEQUENCE IS			*
		4450	*	B CUPFLT			*
		4451	*	SUBJECT TO INPUT CONDITIONS DESCRIBED BELOW.			*
		4452	*	* ENTRY POINT CUPFLT MAY ALSO BE SPECIFIED AS I\$CUPF WHEN CALLED			*
		4453	*	FROM ONE OF THE SUBROUTINES RESIDENT IN VIRTUAL MEMORY.			*
		4454	*				*
		4455	*	*INPUT			*
		4456	*	REGISTER @XR - FOR THE UNPACKED FLOATING POINT VALUE POINTER.			*
		4457	*	THIS CONTAINS THE CORE ADDRESS OF THE LEFTMOST BYTE OF THE			*
		4458	*	FLOATING POINT VALUE TO BE PACKED.			*
		4459	*				*
		4460	*	*OUTPUT			*
		4461	*	PACKED FLOATING POINT VALUE - LOCATED WITH LEFTMOST BYTE			*
		4462	*	REFERENCED BY REGISTER @XR, REPLACING THE ORIGINAL UNPACK			*
		4463	*	FLOATING POINT VALUE.			*
		4464	*				*
		4465	*	*EXTERNAL REFERENCES			*
		4466	*	N/A			*
		4467	*				*
		4468	*	*EXITS, NORMAL			*
		4469	*	CONTROL IS ALWAYS RETURNED TO THE FIRST INSTRUCTION FOLLOWING THE			*
		4470	*	CUPFLT CALLING SEQUENCE.			*
		4471	*				*
		4472	*	*EXITS, ERROR			*
		4473	*	N/A			*
		4474	*				*
		4475	*	*TABLES/WORK AREAS			*
		4476	*	CONVERSION REQUIRES A TEMPORARY WORK AREA ADJACENT TO THE ORIGI-			*
		4477	*	NAL UNPACKED FLOATING POINT VALUE. REGION DISPLACEMENTS (RELA-			*
		4478	*	TIVE TO REGISTER @XR) FOR EACH STEP IN THE CONVERSION ARE -			*
		4479	*	* ORIGINAL PACKED VALUE STD - 0 TO 7, LONG - 0 TO 15			*
		4480	*	* TEMPORARY WORK AREA STD - 8, LONG - 16			*
		4481	*	* FINAL UNPACKED VALUE STD - 0 TO 4, LONG - 0 TO 8			*
		4482	*	THE ORIGINAL CONTENTS OF THE TEMPORARY WORK AREA ARE NOT SAVED.			*
		4483	*				*
		4484	*	*ATTRIBUTES			*

## S/3 BASIC INTERPRETER FLOATING POINT PACKER

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20	PAGE 33
			4485	*	* REUSABLE			*
			4486	*	* RELOCATABLE			*
			4487	*				*
			4488	*	*CHARACTER CODE DEPENDENCY			*
			4489	*	THE OPERATION OF THIS MODULE DEPENDS UPON THE FOLLOWING PROPER-			*
			4490	*	TIES OF THE INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET			*
			4491	*	* MOST CODING HAS BEEN ARRANGED SO THAT REDEFINITION OF CHAR-			*
			4492	*	ACTER CONSTANTS, BY REASSEMBLY, WILL RESULT IN A CORRECT			*
			4493	*	MODULE FOR THE NEW DEFINITION.			*
			4494	*	* NUMERIC CHARACTERS 0 THROUGH 9 ARE PRESUMED TO BE CODED SUCH			*
			4495	*	THAT THE HIGH ORDER FOUR BITS CONTAIN A SIGN ZONE WITH X'F'			*
			4496	*	DEFINING A POSITIVE DIGIT AND X'D' DEFINING A NEGATIVE DIGIT.			*
			4497	*	* DECIMAL NUMBERS MUST ALSO BE CODED SO THAT THE LOW ORDER			*
			4498	*	FOUR BITS, WHEN CONSIDERED AS A BINARY INTEGER, IDENTIFY THE			*
			4499	*	VALUE OF THE DIGIT.			*
			4500	*	THE SPECIFIC INSTRUCTIONS (INSTRUCTION SEQUENCES) WHICH REQUIRE			*
			4501	*	MODIFICATION IF THESE PROPERTIES OF THE CHARACTER SET ARE CHANGED			*
			4502	*	BY BE IDENTIFIED BY -			*
			4503	*	* THE 7 INSTRUCTIONS BEGINNING AT LABEL CUP030.			*
			4504	*	* THE 7 INSTRUCTIONS BEGINNING AT LABEL CUP050.			*
			4505	*	* THE 8 INSTRUCTIONS BEGINNING AT LABEL CUP070.			*
			4506	*				*
			4507	*	*NOTES			*
			4508	*	ERROR PROCEDURES			*
			4509	*	N/A			*
			4510	*				*
			4511	*	REGISTER USAGE			*
			4512	*	* REGISTER @BR IS NOT USED.			*
			4513	*	* REGISTER @XR IS USED AS AN INPUT PARAMETER, AND RETAINS THIS			*
			4514	*	SAME VALUE AT CUPFLT EXIT.			*
			4515	*				*
			4516	*	SAVED/RESTORED AREAS			*
			4517	*	N/A			*
			4518	*				*
			4519	*	MODIFICATION CONSIDERATIONS			*
			4520	*	CUPFLT UTILIZES SEQUENTIAL INSTRUCTIONS RATHER THAN A LOOPING			*
			4521	*	TECHNIQUE FOR MANTISSA PACKING. THE ORDER AND SEQUENCE OF			*
			4522	*	THESE INSTRUCTIONS THEREFORE DEPENDS ON THE NUMBER OF SIGNIFI-			*
			4523	*	CANT DIGITS IN THE MANTISSA.			*
			4524	*				*
			4525	*	REQUIRED MODULES			*
			4526	*	* @SYSEQ - COMMON SYSTEM EQUATES.			*
			4527	*	* @B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.			*
			4528	*	* \$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD. PREC. ONLY).			*
			4529	*	* \$I@LEQ - INTERPRETER PARAMETER EQUATES (FOR LONG PREC. ONLY).			*
			4530	*				*
			4531	*	OTHER			*
			4532	*	N/A			*
			4533	*	*****			*

## S/3 BASIC INTERPRETER FLOATING POINT PACKER

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 34
			4535		*****	
			4536	*	FLOATING POINT VALUE PACKING ROUTINE ENTRY POINT	
			4537		*****	
			4538	*		
			4539	*	ENTER CUPFLT - SAVE THE RETURN ADDRESS	
			4540	*		
0A85	34	08 0AE2	0A85	4541	CUPFLT EQU * CUPFLT ENTRY POINT	
				4542	ST CUP080+@OP1,@ARR SET RETURN BRANCH ADDRESS	
				4543	*	
				4544	* REGISTER @XR CONTAINS THE ADDRESS OF THE FLOATING POINT VALUE TO BE	
				4545	* UNPACKED - MOVE THE PACKED VALUE TO A SAVE AREA IN THE STACK	
				4546	*	
0A89	AC	00 08 00		4547	CUP010 MVC CUPSAV(,@XR),I@UEXP(1,@XR) SAVE UNPACKED VALUE EXPONENT	
				4548	*	
				4549	* SET PACKED VALUE STATUS FOR POSITIVE, STANDARD PRECISION.	
				4550	*	
0A8D	BB	F0 00		4551	CUP020 SBF I@STAT(,@XR),B@TRAC+B@DTYP+B@PREC+B@SIGN INITLZ STATUS	
				4552	*	
				4553	* PERFORM MANTISSA PACKING FOR STANDARD PRECISION DIGITS	
				4554	*	
0A90	A8	03 00 01		4555	CUP030 MNN CUPP01(,@XR),CUPU01(,@XR) PACK HIGH ORDER MANTISSA DIG	
0A94	A8	01 01 02		4556	MZN CUPP02(,@XR),CUPU02(,@XR) PACK 2ND HIGH MANTISSA DIGIT	
0A98	A8	03 01 03		4557	MNN CUPP03(,@XR),CUPU03(,@XR) PACK 3RD HIGH MANTISSA DIGIT	
0A9C	A8	01 02 04		4558	MZN CUPP04(,@XR),CUPU04(,@XR) PACK 4TH HIGH MANTISSA DIGIT	
0AA0	A8	03 02 05		4559	MNN CUPP05(,@XR),CUPU05(,@XR) PACK 5TH HIGH MANTISSA DIGIT	
0AA4	A8	01 03 06		4560	MZN CUPP06(,@XR),CUPU06(,@XR) PACK 6TH HIGH MANTISSA DIGIT	
0AA8	A8	03 03 07		4561	MNN CUPP07(,@XR),CUPU07(,@XR) PACK 7TH HIGH MANTISSA DIGIT	
				4562	*	
				4563	* TEST FOR EXECUTION PRECISION MODE	
				4564	*	
0AAC	F2	87 23		4565	CUP040 JC CUP060,I@PRSW BRANCH IF STANDARD PRECISION	
0AAF	BA	20 00		4566	SBN I@STAT(,@XR),B@PREC SET PACKED STATUS FOR LONG PREC	
				4567	*	
				4568	* PERFORM MANTISSA PACKING FOR LONG PRECISION DIGITS	
				4569	*	
0AB2	A8	01 04 08		4570	CUP050 MZN CUPP08(,@XR),CUPU08(,@XR) PACK 8TH HIGH MANTISSA DIGIT	
0AB6	A8	03 04 09		4571	MNN CUPP09(,@XR),CUPU09(,@XR) PACK 9TH HIGH MANTISSA DIGIT	
0ABA	A8	01 05 0A		4572	MZN CUPP10(,@XR),CUPU10(,@XR) PACK 10TH HIGH MANTISSA DIGIT	
0ABE	A8	03 05 0B		4573	MNN CUPP11(,@XR),CUPU11(,@XR) PACK 11TH HIGH MANTISSA DIGIT	
0AC2	A8	01 06 0C		4574	MZN CUPP12(,@XR),CUPU12(,@XR) PACK 12TH HIGH MANTISSA DIGIT	
0AC6	A8	03 06 0D		4575	MNN CUPP13(,@XR),CUPU13(,@XR) PACK 13TH HIGH MANTISSA DIGIT	
0ACA	A8	01 07 0E		4576	MZN CUPP14(,@XR),CUPU14(,@XR) PACK 14TH HIGH MANTISSA DIGIT	
0ACE	A8	03 07 0F		4577	MNN CUPP15(,@XR),CUPU15(,@XR) PACK 15TH HIGH MANTISSA DIGIT	
				4578	*	
				4579	* ESTABLISH EXPONENT FOR THE PACKED FLOATING POINT VALUE	
				4580	*	
0AD2	AC	00 04 08		4581	CUP060 MVC I@PEXP(,@XR),CUPSAV(1,@XR) MOVE EXP INTO PACKED VALUE	
				4582	*	
				4583	* ESTABLISH THE SIGN STATUS FOR PACKED VALUE	
				4584	*	
0AD6	B8	F0 07		4585	CUP070 TBN I@SIGN(,@XR),B@ZPOS IF UNPACKED SIGN IS POSITIVE	
0AD9	F2	10 03		4586	JT CUP080 * GO EXIT THE PACKING ROUTINE	
0ADC	BA	10 00		4587	SBN I@STAT(,@XR),B@SIGN * ELSE SET NEGATIVE SIGN STATUS	
				4588	*	
				4589	* RETURN CONTROL TO THE CALLING PROGRAM	
				4590	*	

VER 15, MOD 00 06/09/20 PAGE 35

0ADF C0 87 0000	4591 CUP080 B	* - *	RETURN TO CALLING PROGRAM
-----------------	---------------	-------	---------------------------

## S/3 BASIC INTERPRETER FLOATING POINT PACKER

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20	PAGE 36
			4593		*****			
			4594		* FLOATING POINT PACKING ROUTINE EQUATES			
			4595		*****			
			4596		*			
			0008	4597	CUPSAV EQU I@LUFV	DISP	FOR	PACKED SAVE AREA
			4598		*			
			0000	4599	CUPP01 EQU I@PMN1+0	DISP	FOR	PACKED 1ST DIGIT
			0001	4600	CUPP02 EQU I@PMN1+1	DISP	FOR	PACKED 2ND DIGIT
			0001	4601	CUPP03 EQU I@PMN1+1	DISP	FOR	PACKED 3RD DIGIT
			0002	4602	CUPP04 EQU I@PMN1+2	DISP	FOR	PACKED 4TH DIGIT
			0002	4603	CUPP05 EQU I@PMN1+2	DISP	FOR	PACKED 9TH DIGIT
			0003	4604	CUPP06 EQU I@PMN1+3	DISP	FOR	PACKED 6TH DIGIT
			0003	4605	CUPP07 EQU I@PMN1+3	DISP	FOR	PACKED 7TH DIGIT
			4606		*			
			0004	4607	CUPP08 EQU I@PMN1+4	DISP	FOR	PACKED 8TH DIGIT
			0004	4608	CUPP09 EQU I@PMN1+4	DISP	FOR	PACKED 9TH DIGIT
			0005	4609	CUPP10 EQU I@PMN1+5	DISP	FOR	PACKED 10TH DIGIT
			0005	4610	CUPP11 EQU I@PMN1+5	DISP	FOR	PACKED 11TH DIGIT
			0006	4611	CUPP12 EQU I@PMN1+6	DISP	FOR	PACKED 12TH DIGIT
			0006	4612	CUPP13 EQU I@PMN1+6	DISP	FOR	PACKED 13TH DIGIT
			0007	4613	CUPP14 EQU I@PMN1+7	DISP	FOR	PACKED 14TH DIGIT
			0007	4614	CUPP15 EQU I@PMN1+7	DISP	FOR	PACKED 15TH DIGIT
			4615		*			
			0001	4616	CUPU01 EQU I@UMN1+0	DISP	FOR	UNPACKED 1ST DIGIT
			0002	4617	CUPU02 EQU I@UMN1+1	DISP	FOR	UNPACKED 2ND DIGIT
			0003	4618	CUPU03 EQU I@UMN1+2	DISP	FOR	UNPACKED 3RD DIGIT
			0004	4619	CUPU04 EQU I@UMN1+3	DISP	FOR	UNPACKED 4TH DIGIT
			0005	4620	CUPU05 EQU I@UMN1+4	DISP	FOR	UNPACKED 5TH DIGIT
			0006	4621	CUPU06 EQU I@UMN1+5	DISP	FOR	UNPACKED 6TH DIGIT
			0007	4622	CUPU07 EQU I@UMN1+6	DISP	FOR	UNPACKED 7TH DIGIT
			4623		*			
			0008	4624	CUPU08 EQU I@UMN1+7	DISP	FOR	UNPACKED 8TH DIGIT
			0009	4625	CUPU09 EQU I@UMN1+8	DISP	FOR	UNPACKED 9TH DIGIT
			000A	4626	CUPU10 EQU I@UMN1+9	DISP	FOR	UNPACKED 10TH DIGIT
			000B	4627	CUPU11 EQU I@UMN1+10	DISP	FOR	UNPACKED 11TH DIGIT
			000C	4628	CUPU12 EQU I@UMN1+11	DISP	FOR	UNPACKED 12TH DIGIT
			000D	4629	CUPU13 EQU I@UMN1+12	DISP	FOR	UNPACKED 13TH DIGIT
			000E	4630	CUPU14 EQU I@UMN1+13	DISP	FOR	UNPACKED 14TH DIGIT
			000F	4631	CUPU15 EQU I@UMN1+14	DISP	FOR	UNPACKED 15TH DIGIT
			4632		*			
			4633		*****			
			4634		*			
			4635		* END OF FLOATING POINT PACKING ROUTINE CODING			
			4636		*			

## S/3 BASIC INTERPRETER FLOAT TO BIN SUBR CONV

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20	PAGE 37
4638				*****			
4639	*			5703-XM1 COPYRIGHT IBM CORP. 1970			*
4640	*			REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
4641	*						*
4642				*****			*
4643				*STATUS			*
4644	*			VERSION 1 MODIFICATION 0			*
4645	*						*
4646				*FUNCTION			*
4647	*			* CAFPBS CONVERTS A STANDARD OR LONG PRECISION UNPACKED FLOATING			*
4648	*			POINT DECIMAL VALUE TO A TWO,-BYTE BINARY NUMBER SUITABLE FOR			*
4649	*			USE AS AN ARRAY SUBSCRIPT OR EXECUTION INDEXING VALUE.			*
4650	*			* AN UNPACKED (ZONED) DECIMAL FLOATING POINT VALUE REFERENCED BY			*
4651	*			REGISTER @XR IS CONVERTED TO A TWO-BYTE BINARY NUMBER. THE			*
4652	*			FLOATING POINT VALUE MUST BE A POSITIVE QUANTITY WHICH IS NOT			*
4653	*			LESS THAN 1 AND IS LESS THAN 10,000. THE RESULTING BINARY NUM-			*
4654	*			BER IS STORED IN THE FIRST TWO BYTES OCCUPIED BY THE ORIGINAL			*
4655	*			VALUE. REGISTER @XR IS NOT MODIFIED DURING EXECUTION.			*
4656	*						*
4657	*						*
4658				*ENTRY POINTS			*
4659	*			* THIS ROUTINE HAS A SINGLE ENTRY POINT - CAFPBS - WHOSE FUNCTION			*
4660	*			IS DEFINED ABOVE. CALLING SEQUENCE IS			*
4661	*			B CAFPBS			*
4662	*			SUBJECT TO INPUT CONDITIONS DESCRIBED BELOW.			*
4663	*			* ENTRY POINT CAFPBS MAY ALSO BE SPECIFIED AS I\$CFBS WHEN CALLED			*
4664	*			FROM ONE OF THE SUBROUTINES RESIDENT IN VIRTUAL MEMORY.			*
4665	*						*
4666				*INPUT			*
4667	*			* REGISTER PXR - FOR THE UNPACKED FLOATING POINT VALUE POINTER.			*
4668	*			THIS CONTAINS THE CORE ADDRESS OF THE LEFTMOST BYTE OF THE			*
4669	*			FLOATING POINT VALUE TO BE CONVERTED.			*
4670	*						*
4671				*OUTPUT			*
4672	*			* BINARY SUBSCRIPT OR EXECUTION INDEX - 2 BYTES, LOCATED WITH			*
4673	*			LEFTMOST BYTE REFERENCED BY REGISTER @XR, REPLACING THE FIRST			*
4674	*			TWO BYTES OF THE ORIGINAL UNPACKED FLOATING POINT VALUE.			*
4675	*			* IZEQRC - 1 BYTE, FOR THE ERROR CONDITION CODE. THIS CONTAINS			*
4676	*			A NULL CODE (MDR) WHEN NO ERROR CONDITION EXISTS, OR AN			*
4677	*			ERROR CODE SPECIFYING THE PARTICULAR ERROR CONDITION DISCOVERED.			*
4678	*						*
4679	*						*
4680				*EXTERNAL REFERENCES			*
4681	*						*
4682	*			* IZERRC - 1 BYTE, FOR THE INTERPRETER EXECUTION ERROR CODE.			*
4683	*						*
4684				*EXITS, NORMAL			*
4685	*			CONTROL IS ALWAYS RETURNED TO THE FIRST INSTRUCTION FOLLOWING THE			*
4686	*			CAFPBS CALLING SEQUENCE.			*
4687	*						*
4688	*						*
4689				*EXIST, ERROR			*
4690	*			CONTROL IS RETURNED TO THE FIRST INSTRUCTION FOLLOWING THE CAFPBS			*
4691	*			CALLING SEQUENCE WITH INTERPRETER PARAMETER IZERRC CONTAINING THE			*
4692	*			APPROPRIATE ERROR MESSAGE CODE.			*
4693	*						*



## S/3 BASIC INTERPRETER FLOAT TO BIN SUBR CONV

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 38
		4694	*		*
		4695	*	*TABLES/WORK AREAS	*
		4696	*	* CAFBNS - 2 BYTES, FOR THE BINARY NUMBER ACCUMULATOR. THE	*
		4697	*	BINARY NUMBER IS GENERATED IN THIS AREA.	*
		4698	*	* CAFBCR - 6 BYTES FOR THE BINARY CONVERSION MULTIPLIERS. THIS	*
		4699	*	AREA IS INITIALIZED TO CONTAIN THREE 2-BYTE BINARY CONSTANTS	*
		4700	*	REPRESENTING 10, 100, AND 1000. THESE CONSTANTS ARE SHIFTED AS	*
		4701	*	REQUIRED DURING CONVERSION SUCH THAT THE APPROPRIATE BINARY	*
		4702	*	POWER OF 10 MAY BE ADDED TO THE ACCUMULATOR.	*
		4703	*		*
		4704	*	*ATTRIBUTES	*
		4705	*	* REUSABLE	*
		4706	*	* RELOCATABLE	*
		4707	*		*
		4708	*	*CHARACTER CODE DEPENDENCY	*
		4709	*	THE OPERATION OF THIS MODULE DEPENDS UPON THE FOLLOWING PROPER-	*
		4710	*	TIE OF THE INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
		4711	*	* MOST CODING HAS BEEN ARRANGED SO TWAT REDEFINITION OF CHAR-	*
		4712	*	ALTER CONSTANTS, BY REASSEMBLY, WILL RESULT IN A CORRECT	*
		4713	*	MODULE FOR THE NEW DEFINITION.	*
		4714	*	* NUMERIC CHARACTERS 0 THROUGH 9 ARE PRESUMED TO BE CODED SUCH	*
		4715	*	THAT THE HIGH ORDER FOUR BITS CONTAIN A SIGN ZONE WITH X'F'	*
		4716	*	DEFINING A POSITIVE DIGIT.	*
		4717	*	* DECIMAL NUMBERS MUST ALSO BE CODED SO THAT THE LOW ORDER	*
		4718	*	FOUR BITS, WHEN CONSIDERED AS A BINARY INTEGER, IDENTIFY THE	*
		4719	*	VALUE OF THE DIGIT.	*
		4720	*	THE SPECIFIC INSTRUCTIONS (INSTRUCTION SEQUENCES) WHICH REQUIRE	*
		4721	*	MODIFICATION IF THESE PROPERTIES OF THE CHARACTER SET ARE CHANGED	*
		4722	*	MAY BE IDENTIFIED BY -	*
		4723	*	* THE 2 INSTRUCTIONS BEGINNING AT LABEL CAF020.	*
		4724	*	* THE SINGLE INSTRUCTION AT LABEL CAF070.	*
		4725	*	COMMENTS ARE PROVIDED TO INDICATE THE CONSIDERATIONS. INVOLVED	*
		4726	*	AND MECHANISMS FOR CHANGING THE CODE.	*
		4727	*		*
		4728	*	*NOTES	*
		4729	*	ERROR PROCEDURES	*
		4730	*	* ERROR 1 - THE VALUE TO BE CONVERTED IS FOUND TO BE NEGATIVE.	*
		4731	*	* ERROR 2 - THE VALUE TO BE CONVERTED IS FOUND TO BE POSITIVE,	*
		4732	*	BUT LESS THAN 1.	*
		4733	*	* ERROR 3 - THE VALUE TO BE CONVERTED IS FOUND TO BE POSITIVE	*
		4734	*	BUT GREATER THAN OR EQUAL TO 10,000.	*
		4735	*	* IN EACH OF THESE CASES, AN ERROR CODE FOR THE MESSAGE	*
		4736	*	'SUBSCRIPT OUT OF RANGE' IS ESTABLISHED IN INTERPRETER PARA-	*
		4737	*	METER IIERRC AND CONTROL IS RETURNED TO THE CALLING PROGRAM.	*
		4738	*	WHEN THIS OCCURS, THE VALUE TO BE CONVERTED MAY ALREADY HAVE	*
		4739	*	BEEN MODIFIED DURING SUBROUTINE EXECUTION.	*
		4740	*		*
		4741	*	REGISTER USAGE	*
		4742	*	* REGISTER @BR IS SAVED, USED FOR CAFPBS BASE ADDRESSABILITY,	*
		4743	*	THEN RESTORED AT CAFPBS EXIT.	*
		4744	*	* REGISTER @XR IS USED AS AN INPUT PARAMETER, AND RETAINS THIS	*
		4745	*	SAME VALUE AL CAFPBS EXIT.	*
		4746	*		*
		4747	*	SAVED/RESTORED AREAS	*
		4748	*	N/A	*
		4749	*		*

S/3 BASIC INTERPRETER FLOAT TO BIN SUBR CONV							
ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20 PAGE 39	
		4750	*	MODIFICATION CONSIDERATIONS		*	
		4751	*	N/A		*	
		4752	*			*	
		4753	*	REQUIRED MODULES		*	
		4754	*			*	
		4755	*	* @SYSEQ - COMMON SYSTEM EQUATES.		*	
		4756	*	* @ERMEQ - SYSTEM ERROR MESSAGE CODE EQUATES.		*	
		4757	*	* \$B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.		*	
		4758	*	* \$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD. PREC. ONLY).		*	
		4759	*	* \$I@LEQ - INTERPRETER PARARETER EQUATES (FOR LONG PREC. ONLY).		*	
		4760	*	* INTERP - INTERPRETER EXECUTIVE ROUTINE.		*	
		4761	*	* IZCOMN - INTERPRETER COMMON ADDRESS REFERENCE EQUATES.		*	
		4762	*			*	
		4763	*	OTHER		*	
		4764	*	N/A		*	
		4765	*****				

S/3 BASIC INTERPRETER FLOAT TO BIN SUBR CONV							
ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20 PAGE 39	
		4750	*	MODIFICATION CONSIDERATIONS		*	
		4751	*	N/A		*	
		4752	*			*	
		4753	*	REQUIRED MODULES		*	
		4754	*			*	
		4755	*	* @SYSEQ - COMMON SYSTEM EQUATES.		*	
		4756	*	* @ERMEQ - SYSTEM ERROR MESSAGE CODE EQUATES.		*	
		4757	*	* \$B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.		*	
		4758	*	* \$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD. PREC. ONLY).		*	
		4759	*	* \$I@LEQ - INTERPRETER PARARETER EQUATES (FOR LONG PREC. ONLY).		*	
		4760	*	* INTERP - INTERPRETER EXECUTIVE ROUTINE.		*	
		4761	*	* IZCOMN - INTERPRETER COMMON ADDRESS REFERENCE EQUATES.		*	
		4762	*			*	
		4763	*	OTHER		*	
		4764	*	N/A		*	
		4765	*****				

S/3 BASIC INTERPRETER FLOAT TO BIN SUBR CONV						
ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20 PAGE 39
		4750	*	MODIFICATION CONSIDERATIONS		*
		4751	*	N/A		*
		4752	*			*
		4753	*	REQUIRED MODULES		*
		4754	*			*
		4755	*	* @SYSEQ - COMMON SYSTEM EQUATES.		*
		4756	*	* @ERMEQ - SYSTEM ERROR MESSAGE CODE EQUATES.		*
		4757	*	* \$B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.		*
		4758	*	* \$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD. PREC. ONLY).		*
		4759	*	* \$I@LEQ - INTERPRETER PARARETER EQUATES (FOR LONG PREC. ONLY).		*
		4760	*	* INTERP - INTERPRETER EXECUTIVE ROUTINE.		*
		4761	*	* IZCOMN - INTERPRETER COMMON ADDRESS REFERENCE EQUATES.		*
		4762	*			*
		4763	*	OTHER		*
		4764	*	N/A		*
		4765	*	*****		*



## S/3 BASIC INTERPRETER FLOAT TO BIN SUBR CONV

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 40
			4767		*****	
			4768		* FLOATING POINT TO BINARY SUBSCRIPT ROUTINE ENTRY POINT	
			4769		*****	
			4770		*	
			4771		* ENTER CAFPBS - SAVE REGISTERS AND SET ADDRESSABILITY	
			4772		*	
		0AE3	4773	CAFPBS EQU *	CAFPBS ENTRY POINT	
		0AEE	4774	USING CAF020,@BR	DEFINE CAMS BASE ADDRESS	
0AE3	34	01	0B3C	4775	ST CAF130+@OP1,@BR	SAVE CALLING PROS BASE AIEG
0AE7	C2	01	0AEE	4776	LA CAF020,@BR	LOAD CAFPBS BASE REGISTER
0AEB	74	08	52	4777	ST CAF140+@OP1(,@BR),@ARR	SET RETURN BRANCH ADDRESS
			4778		*	
			4779		* REGISTER @XR CONTAINS THE ADDRESS OF NE FLOATING POINT VALUE TO BE	
			4780		* CONVERTED - TEST FOR A NEGATIVE VALUE (AN ERROR CONDITION)	
			4781		*	
0AEE	B8	F0	07	4782	CAF020 TBN I@SIGN(,@XR),B@ZPOS	IF STACKED VALUE IS NEGATIVE
0AF1	F2	90	0D	4783	JF CAF040	* GO EXIT ON ERROR CONDITION
			4784		*	
			4785		* TEST THE MAGNITUDE OR THE FLOATING VALUE - VALUES TO BE CONVERTED	
			4786		* MUST BE GREATER THAN OR EQUAL TO 1 AND LESS THAN 10,000	
			4787		*	
0AF4	BD	84	00	4788	CAF030 CLI I@UEXP(,@XR),B@NXZR+B@LDDM	IF EXPONENT GREATER THAN E+04
0AF7	F2	84	07	4789	JH CAF040	* GO EXIT ON ERROR CONDITION
			4790		*	
0AFA	9F	00	00 55	4791	SLC I@UEXP(,@XR),CAFNXZ(1,@BR)	MAKE EXPONENT A DISPLACEMENT
0AFE	F2	84	07	4792	JH CAF050	IF EXPONENT GREATER THAN E+00
			4793		*	* GO CONTINUE THE CONVERSION
			4794		*	
			4795		* ERROR EXIT - SET ERROR ROUTINE TO DISPLAY 'SUBSCRIPT OUT OF RANGE'	
			4796		*	
0B01	3C	D1	0CBC	4797	CAF040 MVI IZERRC,@E760	SET NE ERROR MESSAGE CODE
0B05	D0	87	4B	4798	B CAF130(,@BR)	GO RETURN TO CALLING PROGRAM
			4799		*	
			4800		* INITIALIZE FOR DECIMAL TO BINARY CONVERSION	
			4801		*	
0B08	6C	00	2D 00	4802	CAF050 MVC CAF070+@DD2(,@BR),I@UEXP(1,@XR)	SET INSTRUCTIONS FOR DISP
0B0C	6C	00	37 00	4803	MVC CAF090+@D1(,@BR),I@UEXP(1,@XR)	* TO UNITS DIGIT IN VALUE
0B10	5C	03	5F 59	4804	MVC CAFBCR(,@BR),CAFBC(2*B@LDMN,@BR)	SET BINARY MULTIPLIERS
			4805		*	* TO CONVERT TENS DIGIT
			4806		*	
			4807		* CONVERT UNITS DIGIT IN VALUE TO A BINARY NUMBER	
			4808		*	
0B14	5F	01	61 61	4809	CAF060 SLC CAFBNS(,@BR),CAFBN(B@LDMN,@BR)	CLEAR BINARY ACCUMULATOR
0B18	68	03	61 00	4810	CAF070 MNN CAFBNS(,@BR),*-(,@XR)	CONVERT UNITS DIGIT TO BINARY
0B1C	F2	87	0F	4811	J CAF110	GO CONVERT REMAINING DIGITS
			4812		*	
			4813		* ADD A BINARY POWER OF 10 TO THE ACCUMULATOR - DO THIS AS MANY TIMES	
			4814		* AS SPECIFIED BY THE DECIMAL DIGIT BEING CONVERTED	
			4815		*	
0B1F	5E	01	61 5F	4816	CAF080 ALC CAFBNS(,@BR),CAFBCR(B@LDMN,@BR)	ADD BINARY POWER OF 10
0B23	97	00	00 54	4817	CAF090 SZ *-(1,@XR),CAFDN1(1,@BR)	DECREMENT THE DECIMAL DIGIT
0B27	D0	02	31	4818	BNL CAF080(,@BR)	* AND REPEAT LOOP UNTIL ZERO
			4819		*	
			4820		* ADJUST THE MULTIPLIER REGISTER FOR NEXT NIGNER ORDER OF 10	
			4821		*	
0B2A	5C	03	5F 5D	4822	CAF100 MVC CAFBCR(,@BR),CAFBCR-2(2*B@LDMN,@BR)	SHIFT MULTIPLIERS

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20	PAGE 41
			4823	*				
			4824	*	ADJUST DECIMAL VALUE POINTER TO NEXT HIGHER ORDER DIGIT			
			4825	*				
0B2E	5F	00 37 53	4826	CAF110 SLC	CAF090+@D1(,@BR),CAFBN1(1,@BR) DECREMENT DIGIT POINTER			
0B32	D0	84 35	4827	BH	CAF090(,@BR) BRANCH IF MORE DIGITS REMAIN			
			4828	*				
			4829	*	MOVE TNE BINARY SUBSCRIPT TO THE RUN-TIME STACK			
			4830	*				
0B35	9C	01 01 61	4831	CAF120 MVC	B@LDMN-1(,@XR),CAFBN5(B@LDMN,@BR) STACK THE SUBSCRIPT			
			4832	*				
			4833	*	NORMAL EXIT - RETURN CONTROL TO CALLING PROGRAM			
			4834	*				
0B39	C2	01 0000	4835	CAF130 LA	*-*,@BR RESTORE CALLING PROGRAM BASE			
0B3D	C0	87 0000	4836	CAF140 B	*-* RETURN TO CALLING PROGRAM			

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 42
		4838		*****	
		4839		* FLOATING TO BINARY SUBSCRIPT ROUTINE CONSTANTS AND WORK AREAS	
		4840		*****	
		4841		*	
0B41 01		0B41 4842	CAFBN1 DC	XL1'1'	BINARY CONSTANT +1
0B42 F1		0B42 4843	CAFDN1 DC	DL1'1'	DECIMAL CONSTANT +1
0B43 80		0B43 4844	CAFNXZ DC	AL1(B@NXZR)	ZERO NORMALIZED EXPONENT
		4845		*	
0B44 0064		0B45 4846	DC	IL(B@LDMN)'100'	10**2 CONVERSION MULTIPLIER
0B46 000A		0B47 4847	CAFBCC DC	IL(B@LDMN)'10'	10**1 CONVERSION MULTIPLIER
		4848		*	
0B48 03E8		0B49 4849	DC	IL(B@LDMN)'1000'	10**3 CONVERSION MULTIPLIER
0B4A		0B4B 4850	DS	CL(B@LDMN)	CONVERSION REGISTER - PENDING
0B4C		0B4D 4851	CAFBCR DS	CL(B@LDMN)	CONVERSION REGISTER - CURRENT
		4852		*	
0B4E		0B4F 4853	CAFBNS DS	CL(B@LDMN)	BINARY SUBSCRIPT ACCUMULATOR
		4854		*	
		4855		* END OF FLOATING POINT TO BINARY SUBSCRIPT ROUTINE CODING	
		4856		*	

## S/3 BASIC INTERPRETER ELEMENT STACKING RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 43
		4858		*****	
		4859	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		4860	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
		4861	*		*
		4862		*****	
		4863		*STATUS	*
		4864	*	VERSION 1 MODIFICATION 0	*
		4865	*		*
		4866		*FUNCTION	*
		4867	*	* ISTACK MOVES VARIABLE LENGTH DATA FIELDS FROM VIRTUAL MEMORY TO	*
		4868	*	ANY GIVEN CORE LOCATION.	*
		4869	*	* A VARIABLE LENGTH DATA FIELD IS MOVED FROM VIRTUAL MEMORY TO	*
		4870	*	THE CORE LOCATION (NORMALLY WITHIN THE RUN-TIME STACK) REFER-	*
		4871	*	ENCED BY REGISTER ONE. THE FIELD IS REFERENCED IN VIRTUAL	*
		4872	*	MEMORY USING PAGING PARAMETER IZVADR, AND MAY EXTEND ACROSS A	*
		4873	*	SINGLE VIRTUAL PAGE BOUNDARY.	*
		4874	*	* FIELD LENGTH IS SPECIFIED IN A ONE-BYTE PARAMETER TO THE SUB-	*
		4875	*	ROUTINE, AND REMAINS AVAILABLE AFTER SUBROUTINE EXECUTION.	*
		4876	*	* REGISTER @XR IS NOT MODIFIED DURING EXECUTION, BUT THE VIRTUAL	*
		4877	*	ADDRESS IN VADDR IS SUBJECT TO MODIFICATION WHEN A PAGE BOUND-	*
		4878	*	ARY CONDITION EXISTS.	*
		4879	*		*
		4880		*ENTRY POINTS	*
		4881	*	* THIS ROUTINE HAS A SINGLE ENTRY POINT - ISTACK - WHOSE FUNCTION	*
		4882	*	IS DEFINED ABOVE. CALLING SEQUENCE IS	*
		4883	*	B ISTACK	*
		4884	*	SUBJECT TO INPUT CONDITIONS DESCRIBED BELOW.	*
		4885	*	* ENTRY POINT ISTACK MAY ALSO BE SPECIFIED AS I\$STCK WHEN CALLED	*
		4886	*	FROM ONE OF THE SUBROUTINES RESIDENT IN VIRTUAL MEMORY.	*
		4887	*		*
		4888		*INPUT	*
		4889	*	* REGISTER @XR - FOR THE DESTINATION CORE LOCATION POINTER. THIS	*
		4890	*	CONTAINS THE CORE ADDRESS OF THE LEFTMOST BYTE OF THE CORE AREA	*
		4891	*	INTO WHICH THE DATA ELEMENT IS TO BE MOVED.	*
		4892	*	* IZVADR - 2 BYTES, FOR THE PAGING ROUTINE VIRTUAL ADDRESS PARA-	*
		4893	*	METER. THIS CONTAINS THE VIRTUAL ADDRESS OF THE LEFTMOST BYTE	*
		4894	*	OF THE DATA ELEMENT WHICH IS TO BE MOVED.	*
		4895	*	* ISTLNG (EXTERNAL IZ\$SLNG, I\$SLNG) - 1 BYTE, FOR THE DATA ELEMENT	*
		4896	*	LENGTH CODE. THIS CONTAINS A VALUE WHICH IS ONE LESS THAN THE	*
		4897	*	ACTUAL LENGTH OF THE DATA ELEMENT. UNLESS SPECIFICALLY SET	*
		4898	*	PRIOR TO SUBROUTINE EXECUTION, I\$TLNG AUTOMATICALLY CONTAINS	*
		4899	*	THE LENGTH CODE REQUIRED TO MOVE A PACKED FLOATING POINT DECI-	*
		4900	*	MAL VALUE (5 BYTES FOR STANDARD PRECISION, 9 BYTES FOR LONG).	*
		4901	*		*
		4902		*OUTPUT	*
		4903	*	* STACKED DATA ELEMENT - THIS ELEMENT, OF LENGTH (ISTLNG+1) BYTES,	*
		4904	*	IS LOCATED WITH LEFTMOST BYTE STORED AT THE ADDRESS SPECIFIED	*
		4905	*	IN REGISTER @XR.	*
		4906	*	* ISTLLC (EXTERNAL IZ\$SLLC, I\$SLLC) 1 BYTE, FOR THE STACKED ELE-	*
		4907	*	MENT LENGTH CODE. THIS CONTAINS A VALUE WHICH IS ONE LESS THAN	*
		4908	*	THE ACTUAL LENGTH OF THE CURRENTLY STACKED ELEMENT.	*
		4909	*		*
		4910		*EXTERNAL REFERENCES	*
		4911	*	* IPGCVA - ENTRY POINT FOR PAGING MODULE VADDR CONVERSION RTN.	*
		4912	*	* IZPGNO - 1 BYTE, FOR PAGING MODULE VIRTUAL PAGE NO. PARAMETER.	*
		4913	*	* IZPGDS - 1 BYTE, FOR PAGING MODULE VIRTUAL PAGE DISP. PARAMETER.	*

## S/3 BASIC INTERPRETER ELEMENT STACKING RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20	PAGE 44
		4914	*	* IZCADR - 2 BYTES, FOR PAGING MODULE CORE ADDRESS OUTPUT PARAM.	*		
		4915	*		*		
		4916	*	*EXITS, NORMAL	*		
		4917	*	CONTROL IS ALWAYS RESTORED TO TSE FIRST INSTRUCTION FOLLOWING THE	*		
		4918	*	ISTACK CALLING SEQUENCE.	*		
		4919	*		*		
		4920	*	*EXITS, ERROR	*		
		4921	*	N/A	*		
		4922	*		*		
		4923	*	*TABLES/WAIAREAS	*		
		4924	*	N/A	*		
		4925	*		*		
		4926	*	*ATTRIBUTES	*		
		4927	*	* REUSABLE	*		
		4928	*	* RELOCATABLE	*		
		4929	*		*		
		4930	*	*CHARACTER CODE DEPENDENCY	*		
		4931	*	THE OPERATION OR THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*		
		4932	*	REPRESENTATION OF NE EXTERNAL CHARACTER SET.	*		
		4933	*		*		
		4934	*	*NOTES	*		
		4935	*	ERROR PROCEDURES	*		
		4936	*	N/A	*		
		4937	*		*		
		4938	*	REGISTER USAGE	*		
		4939	*	* REGISTER @BR IS SAVED, USED FOR ISTACK BASE ADDRESSABILITY,	*		
		4940	*	THEN RESTORED AT ISTACK EXIT.	*		
		4941	*	* REGISTER @XR IS USED AS AN INPUT PARAMETER, AND RETAINS THIS	*		
		4942	*	SAME VALUE AT ISTACK EXIT.	*		
		4943	*		*		
		4944	*	SAVED/RESTORED AREAS	*		
		4945	*	N/A	*		
		4946	*		*		
		4947	*	MODIFICATION CONSIDERATIONS	*		
		4948	*	N/A	*		
		4949	*		*		
		4950	*	REQUIRED MODULES	*		
		4951	*	* @SYSEQ - COMMON SYSTEM EQUATES.	*		
		4952	*	* \$B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.	*		
		4953	*	* \$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD. PREC. ONLY).	*		
		4954	*	* \$I@LEQ - INTERPRETER PARAMETER EQUATES (FOR LONG PREC. ONLY).	*		
		4955	*	* IPGMDL - INTERPRETER PAGING CONTROL MODULE.	*		
		4956	*	* IZCOMN - INTERPRETER COMMON ADDRESS REFERENCE EQUATES.	*		
		4957	*		*		
		4958	*	OTHER	*		
		4959	*	N/A	*		
		4960	*	*****	*		

## S/3 BASIC INTERPRETER ELEMENT STACKING RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 45
		4962		*****	
		4963		* ELEMENT STACKING ROUTINE ENTRY POINT	
		4964		*****	
		4965		*	
		4966		* ENTER ISTACK - SAVE REGISTERS AND SET ADDRESSABILITY	
		4967		*	
		0B50 4968	ISTACK EQU *	ISTACK ENTRY POINT	
		0B5B 4969	USING IST010,@BR	DEFINE ISTACK BASE ADDRESS	
0B50 34 01 0BAA		4970	ST IST120+@OP1,@BR	SAVE CALLING PROG BASE REG	
0B54 C2 01 0B5B		4971	LA IST010,@BR	LOAD ISTACK BASE REGISTER	
0B58 74 08 53		4972	ST IST130+@OP1(,@BR),@ARR	SET RETURN BRANCH ADDRESS	
		4973	*		
		4974		* COMPLETE THE FINAL VM/STACK MOVE INSTRUCTION - THE 2ND DISPLACEMENT	
		4975		* IN THIS INSTRUCTION CONTAINS THE ELEMENT LENGTH CODE INPUT PARAMETER	
		4976		*	
0B5B 5C 01 46 47		4977	IST010 MVC IST100+@DD2-1(,@BR),IST100+@DD2(@INST4-2,@BR)	PROPAGATE	
		4978	*	* THE ELEMENT LENGTH PARAMETER	
		4979	*		
		4980		* TEST FOR A POSSIBLE VIRTUAL PAGE BOUNDARY CONDITION	
		4981	*		
0B5F 4C 00 2D 144A		4982	IST020 MVC IST070+@Q(,@BR),IZPGDS(@VADDR-1)	COMPUTE 2ND SEGMENT	
0B64 5E 00 2D 47		4983	ALC IST070+@Q(,@BR),IST100+@DD2(1,@BR)	* LENGTH PARAMETER -	
		4984	*	* IF ELEMENT RESIDES ON SINGLE	
0B68 F2 82 2C		4985	JL IST080	* PAGE. GO STACK SINGLE ELEMENT	
		4986	*		
		4987		*****	



## S/3 BASIC INTERPRETER ELEMENT STACKING RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 46
		4989		*****	
		4990		* PAGE BOUNDARY CONDITION - ACCESS PAGE CONTAININS FIRST SEGMENT	
		4991		*****	
		4992		*	
0B6B	C0 87 1358	4993	IST030 B	IPGCVA LINK TO GET PAGE & CVRT VADDR	
		4994		*	
		4995		* ESTABLISH THE ELEMENT FIRST SEGMENT STACKING INSTRUCTION	
		4996		*	
0B6F	7C FF 27	4997	IST040 MVI	IST060+@DD2(,@BR),B@LVPG-1 CALCULATE 1ST SEGMENT	
0B72	4F 00 27 144A	4998	SLC	IST060+@DD2(,@BR),IZPGDS(@VADDR-1) * LENGTH PARAMETER	
0B77	5C 01 26 27	4999	MVC	IST060+@DD2-1(,@BR),IST060+@DD2(@INST4-2,@BR) PROPAGATE	
		5000		* 1ST SEGMENT LENGTH PARAMETER	
		5001		*	
		5002		* STACK THE FIRST SEGMENT OF ELEMENT RESIDING ON 2 PAGES	
		5003		*	
0B7B	35 01 144C	5004	IST050 L	IZCADR,@BR LOAD THE FIRST SEGMENT CADDR	
0B7F	9C 00 00 00	5005	IST060 MVC	*-*(,@XR),*-*(@VQ,@BR) MOVE 1ST SEGMENT TO THE STACK	
0B83	C2 01 0B5B	5006	LA	IST010,@BR RESTORE ISTACK BASE ADDRESS	
		5007		*	
		5008		* ESTABLISH CONDITIONS TO STACK THE ELEMENT SECOND SEGMENT	
		5009		*	
0B87	7C 00 47	5010	IST070 MVI	IST100+@DD2(,@BR),*-* SET STACKING INST DISP	
0B8A	5C 00 45 47	5011	MVC	IST100+@Q(,@BR),IST100+@DD2(1,@BR) * AND LENGTH FIELDS	
		5012		*	
0B8E	1E 00 1449 54	5013	ALC	IZPGNO,ISTBN1(@VADDR-1,@BR) ADJUST PAGING ROUTINE PARAM	
0B93	3C 00 144A	5014	MVI	IZPGDS,@ZERO * TO REFERENCE NEXT V.M. PAGE	
		5015		*	
		5016		*****	

## S/3 BASIC INTERPRETER ELEMENT STACKING RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 47
		5018		*****	
		5019		* ACCESS VIRTUAL PAGE CONTAINING ELEMENT FINAL (SECOND OR ONLY) SEGMENT	
		5020		*****	
		5021		*	
0B97	C0 87 1358	5022	IST080 B	IPGCVA LINK TO GET PAGE @ CVRT VADDR	
		5023		*	
		5024		* STACK THE ELEMENT FINAL (SECOND OR ONLY) SEGMENT	
		5025		*	
0B9B	35 01 144C	5026	IST090 L	IZCADR,@BR LOAD THE FINAL SEGMENT CADDR	
		5027		*	
0B9F	9C 00 00 00	5028	IST100 MVC	*-*(,@XR),*-*(@VQ,@BR) MOVE FINAL SEGMENT TO THE STACK	
0BA2		5029	ORG	IST100+@DD2 INITIALIZE THE FINAL MOVE INST	
0BA2	04	0BA2 5030	DC	AL1(I@LPFV-1) * TO STACK CURRENT PRECISION	
0BA3		5031	ORG	IST100+@INST4 * ARITHMETIC ELEMENT	
		5032		*	
0BA3	3C 04 0BA2	5033	IST110 MVI	IST100+@DD2,I@LPFV-1 RESET THE ELEMENT LENGTH CODE	
		5034		*	
		5035		*	
		5036		* EXIT - RETURN CONTROL TO THE CALLING PROGRAM	
		5037		*	
0BA7	C2 01 0000	5038	IST120 LA	*-*,@BR RESTORE CALLING PROGRAM BASE	
0BAB	C0 87 0000	5039	IST130 B	*-* RETURN TO CALLING PROGRAM	
		5040		*	
		5041		*****	

S/3 BASIC INTERPRETER ELEMENT STACKING RTN

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 06/09/20 PAGE 48

5043 \*\*\*\*\*  
5044 \* ELEMENT STACKING ROUTINE CONSTANTS  
5045 \*\*\*\*\*  
5046 \*  
0BAF 01 0BAF 5047 ISTBN1 DC IL1'1' BINARY INTEGER +1  
5048 \*  
5049 \*\*\*\*\*  
5050 \* ELEMENT STACKING ROUTINE EQUATES REFERENCING PROGRAM  
5051 \*\*\*\*\*  
5052 \*  
0BA2 5053 ISTLNG EQU IST100+@DD2 ELEMENT LENGTH CODE INPUT PARAM  
5054 \* \* (ELEMENT LENGTH - 1)  
0BA1 5055 ISTLLC EQU IST100+@D1 LAST STACKED ELEMENT LENGTH  
5056 \* \* CODE (ELEMENT LENGTH - 1)  
5057 \*  
5058 \* END OF ELEMENT STACKING ROUTINE CODING  
5059 \*

## S/3 BASIC INTERPRETER ELEMENT STACKING RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20	PAGE 49
		5061		*****			
		5062	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
		5063	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
		5064	*				*
		5065		*****			*
		5066	*	*STATUS			*
		5067	*	VERSION 1 MODIFICATION 0			*
		5068	*				*
		5069	*	*FUNCTION			*
		5070	*	* IUSTAK MOVES VARIABLE LENGTH DATA FIELDS FROM ANY GIVEN CORE			*
		5071	*	LOCATION TO VIRTUAL MEMORY. SOURCE FIELDS MAY BE MATCHED WITH			*
		5072	*	DESTINATION FIELDS TO ENSURE TRANSFER OF CONSISTENT DATA ELE-			*
		5073	*	MENT TYPES, AND DESTINATION FIELDS WHICH SPECIFY ELEMENT			*
		5074	*	TRACING CAN CAUSE THE NEW VALUES TO BE DISPLAYED.			*
		5075	*	* A VARIABLE LENGTH DATA FIELD IS MOVED FROM THE CORE LOCATION			*
		5076	*	(NORMALLY WITHIN THE RUN-TIME STACK) REFERENCED BY REGISTER @XR			*
		5077	*	TO VIRTUAL MEMORY. THE DESTINATION FIELD IS REFERENCED IN			*
		5078	*	VIRTUAL MEMORY USING PAGING PARAMETER IZVADR, AND MAY EXTEND			*
		5079	*	ACROSS A SINGLE PAGE BOUNDARY.			*
		5080	*	* FIELD LENGTH IS SPECIFIED IN A ONE-BYTE PARAMETER TO THE SUB-			*
		5081	*	ROUTINE.			*
		5082	*	* REGISTER @XR IS RETURNED TO THE CALLING PROGRAM INTACT, BUT THE			*
		5083	*	VIRTUAL ADDRESS IN IZVADR IS SUBJECT TO MODIFICATION WHEN A			*
		5084	*	PAGE BOUNDARY CONDITION EXISTS.			*
		5085	*	* DEPENDING ON A SUBROUTINE PARAMETER SETTING, THE SOURCE DATA			*
		5086	*	TYPE MAY BE COMPARED WITH THE DATA TYPE CONTAINED IN THE DESTI-			*
		5087	*	NATION FIELD (ARITHMETIC OR CHARACTER). INCONSISTENT DATA			*
		5088	*	TYPES CAUSE EXECUTION TO BE ABORTED ON AN ERROR CONDITION.			*
		5089	*	* ALSO, DEPENDING ON THE CURRENT EXECUTION MODE OF THE SYSTEM,			*
		5090	*	THE NEW VALUE OF AN ELEMENT WHOSE DESTINATION FIELD IS FLAGGED			*
		5091	*	FOR TRACING IS DISPLAYED ON THE SYSTEM OUTPLT DEVICE.			*
		5092	*				*
		5093	*	*ENTRY POINTS			*
		5094	*	* THIS ROUTINE HAS A SINGLE ENTRY POINT - IUSTAK- WHOSE FUNCTION			*
		5095	*	IS DEFINED ABOVE. CALLING SEQUENCE IS			*
		5096	*	B IUSTAK			*
		5097	*	SUBJECT TO INPUT CONDITIONS DESCRIBED BELOW.			*
		5098	*	* ENTRY POINT IUSTAK MAY ALSO BE SPECIFIED AS I\$USTK WHEN CALLED			*
		5099	*	FROM ONE OF THE SUBROUTINES RESIDENT IN VIRTUAL MEMORY.			*
		5100	*				*
		5101	*	*INPUT			*
		5102	*	* REGISTER @XR - FOR THE SOURCE CORE LOCATION POINTER. THIS CON-			*
		5103	*	TAINS THE CORE ADDRESS OF THE LEFTMOST BYTE OF THE CORE AREA			*
		5104	*	FROM WHICH THE DATA ELEMENT IS TO BE MOVED.			*
		5105	*	* IZVADR - 2 BYTES, FOR THE PAGING ROUTINE VIRTUAL ADDRESS PARA-			*
		5106	*	METER. THIS CONTAINS THE VIRTUAL ADDRESS OF THE LEFTMOST BYTE			*
		5107	*	OF THE DESTINATION FIELD IN VIRTUAL MEMORY.			*
		5108	*	* IUSLNG (EXTERNAL IZULNG I\$ULNG) - 1 BYTE, FOR THE DATA ELEMENT			*
		5109	*	LENGTH CODE. THIS CONTAINS A VALUE WHICH IS 1 LESS THAN THE			*
		5110	*	ACTUAL LENGTH OF THE DATA ELEMENT. UNLESS SPECIFICALLY SET			*
		5111	*	PRIOR TO SUBROUTINE EXECUTION, IUSLNG AUTOMATICALLY CONTAINS			*
		5112	*	THE LENGTH CODE REQUIRED TO MOVE A PACKED FLOATING POINT DECI-			*
		5113	*	MAL VALUE (5 BYTES FOR STANDARD PRECISION, 9 BYTES FOR LONG).			*
		5114	*	* IUSDSW (EXTERNAL IZDMSW,I\$DMSW) - 1 BYTE, FOR THE UNSTACKING			*
		5115	*	ROUTINE DATA MATCHING SWITCH. THIS CONTAINS CODE @NOP WHEN			*
		5116	*	MATCHING IS TO BE PERFORMED, OR CODE @UCB WHEN MATCHING IS NOT			*

## S/3 BASIC INTERPRETER ELEMENT STACKING RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 50
		5117	*	REQUIRED.	*
		5118	*	* \$XIND1 - 1 BYTE, FOR SYSTEM EXECUTION INDICATOR 1. THIS INDI-	*
		5119	*	CATOR CONTAINS A BIT (MASK \$TRACE) WHICH IS SET TO '1' WHEN	*
		5120	*	'TRACE' MODE EXECUTION HAS BEEN SPECIFIED.	*
		5121	*		*
		5122	*	OUTPUT	*
		5123	*	* UNSTACKED DATA ELEMENT - THIS ELEMENT, OF LENGTH (IUSLNG+1)	*
		5124	*	BYTES, IS LOCATED WITH LEFTMOST BYTE STORED IN VIRTUAL MEMORY	*
		5125	*	AT THE ADDRESS ORIGINALLY SPECIFIED IN IZVADR.	*
		5126	*	* TRACED VARIABLE - WHEN 'TRACE' MODE HAS BEEN SPECIFIED AND THE	*
		5127	*	DESTINATION FIELD HAS BEEN FLAGGED FOR VARIABLE TRACE, THE	*
		5128	*	UNSTACKED VALUE IS DISPLAYED, IN ASSOCIATION WITH THE BASIC	*
		5129	*	IDENTIFIER CORRESPONDING TO THE DESTINATION FIELD, ON THE	*
		5130	*	SYSTEM PRINT DEVICE.	*
		5131	*	* IZERRC - 1 BYTE, FOR THE ERROR CONDITION CODE. THIS CONTAINS	*
		5132	*	A NULL CODE (I@NERR) WHEN NO ERROR CONDITION EXISTS, OR AN	*
		5133	*	ERROR CODE SPECIFYING THE PARTICULAR ERROR CONDITION DISCOVERED.	*
		5134	*		*
		5135	*	EXTERNAL REFERENCES	*
		5136	*	* IPGMOD - ENTRY POINT FOR PAGING MODULE V.M. PAGE MODIFY ROUTINE.	*
		5137	*	* IPGCAL - ENTRY POINT FOR PAGING MODULE V.M. PAGE CALL ROUTINE.	*
		5138	*	* V\$DTVR - VIRTUAL ENTRY ADDRESS FOR FZVART, VARIABLE TRACE QIN.	*
		5139	*	* INTERR - ENTRY POINT FOR INTERPRETER EXECUTION ERGCR ROUTINE.	*
		5140	*	* IZVADR - 2 BYTES, FOR PAGING MODULE VIRTUAL ADDRESS INPUT PARAM.	*
		5141	*	* IZPGNO - 1 BYTE, FOR PAGING MODULE VIRTUAL PAGE NO. PARAMETER.	*
		5142	*	* IZPGDS - 1 BYTE, FOR PAGING MODULE VIRTUAL PAGE INSP. PARAMETER.	*
		5143	*	* IZCADR - 2 BYTE, FOR PAGING MODULE CORE ADDRESS OUTPUT PARAM.	*
		5144	*	* IZPARM - 2 BYTES, FOR INTERPRETER COMMUNICATION PARAMETER.	*
		5145	*	THIS IS USED IN IUSTAK TO PASS A VIRTUAL ADDRESS TO VIRTUAL	*
		5146	*	MEMORY RESIDENT TRACE ROUTINE FZVART.	*
		5147	*	* IZERRC - 1 BYTE, FOR THE INTERPRETER EXECUTION ERROR CODE.	*
		5148	*	* \$XIND1 - 1 BYTE, FOR SYSTEM EXECUTION INDICATOR 1.	*
		5149	*		*
		5150	*	EXITS, NORMAL	*
		5151	*	CONTROL IS NORMALLY RETURNED TO THE FIRST INSTRUCTION FOLLOWING	*
		5152	*	THE IUSTAK CALLING SEQUENCE.	*
		5153	*		*
		5154	*	EXITS, ERROR	*
		5155	*	CONTROL IS PASSED TO THE INTERPRETER EXECUTIVE AT ENTRY POINT	*
		5156	*	INTERR WITH PARAMETER IZERRC CONTAINING THE APPROPRIATE ERROR	*
		5157	*	MESSAGE CODE (SEE ERROR PROCEDURES).	*
		5158	*		*
		5159	*	TABLES/WORK AREAS	*
		5160	*	N/A	*
		5161	*		*
		5162	*	ATTRIBUTES	*
		5163	*	* REUSABLE	*
		5164	*	* RELOCATABLE	*
		5165	*		*
		5166	*	CHARACTER CODE DEPENDENCY	*
		5167	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*
		5168	*	REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
		5169	*		*
		5170	*	NOTES	*
		5171	*	ERROR DROCEDURES	*
		5172	*	* ERROR 1 - THE SUBROUTINE IS CONDITIONED TO COMPARE DATA	*

## S/3 BASIC INTERPRETER ELEMENT STACKING RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 51
		5173	*	TYPES, AND AN ATTEMPT IS MADE TO UNSTACK AN ARITHMETIC	*
		5174	*	ELEMENT TO A VIRTUAL MEMORY LOCATION CONTAINING A CHARACTER	*
		5175	*	ELEMENT.	*
		5176	*	* ERROR 2 - THE SUBROUTINE IS CONDITIONED TO COMPARE DATA	*
		5177	*	TYPES, AND AN ATTEMPT IS MADE TO UNSTACK AN CHARACTER	*
		5178	*	ELEMENT TO A VIRTUAL MEMORY LOCATION CONTAINING AN ARITHMETIC	*
		5179	*	ELEMENT.	*
		5180	*	* IN EACH OF THESE CASES, AN ERROR CODE FOR THE MESSAGE	*
		5181	*	'INVALID VARIABLE ASSIGNMENT' IS ESTABLISHED IN INTERPRETER	*
		5182	*	PARAMETER IZERRC, AND CONTROL IS PASSED TO INTERPRETER	*
		5183	*	ERROR ROUTINE INTERR.	*
		5184	*		*
		5185	*	REGISTER USAGE	*
		5186	*	* REGISTER (@BR IS SAVED, USED FOR GENERAL PURPOSE INDEXING,	*
		5187	*	THEN RESTORED AT IUSTAK EXIT,	*
		5188	*	* REGISTER @XR IS USED AS AN INPUT PARAMETER, AND RETAINS THIS	*
		5189	*	SAME VALUE AT IUSTAK EMT UNLESS VARIABLE TRACE IS IN EFFECT.	*
		5190	*		*
		5191	*	SAVED/RESTORED AREAS	*
		5192	*	N/A	*
		5193	*		*
		5194	*	MODIFICATION CONSIDERATIONS	*
		5195	*	N/A	*
		5196	*		*
		5197	*	REQUIRED MODULES	*
		5198	*	* @SYSEQ - COMMON SYSTEM EQUATES.	*
		5199	*	* @FXDEQ - SYSTEM NUCLEUS ADDRESSES AND INDICATOR EQUATES.	*
		5200	*	* @ERMEQ - SYSTEM ERROR MESSAGE CODE EQUATES.	*
		5201	*	* \$V\$EQU - VIRTUAL MEMORY FIXED ADDRESS EQUATES.	*
		5202	*	* \$B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.	*
		5203	*	* \$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD. REC. ONLY).	*
		5204	*	* \$I@LE0 - INTERPRETER PARAMETER EQUATES (FOR LONG REC. ONLY).	*
		5205	*	* INTERP - INTERPRETER EXECUTIVE ROUTINE.	*
		5206	*	* IPGMDL - INTERPRETER PAGING CONTROL MODULE.	*
		5207	*	* IZCOMN - INTERPRETER COMMON ADDRESS REFERENCE EQUATES.	*
		5208	*		*
		5209	*	OTHER	*
		5210	*	N/A	*
		5211	*	*****	*



## S/3 BASIC INTERPRETER ELEMENT STACKING RTN

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 06/09/20 PAGE 52

```

5213 *****
5214 * ELEMENT UNSTACKING ROUTINE ENTRY POINT
5215 *****
5216 *
5217 * ENTER IUSTAK - SAVE REGISTERS AND SET ADDRESSAFLITY
5218 *
0BB0 5219 IUSTAK EQU * IUSTAK ENTRY POINT
0BB8 5220 USING IUS010,@BR DEFINE ILISTAK BASE ADDRESS
0BB0 34 01 0C42 5221 ST IUS140+@OP1,@BR SAVE CALLING PROG BASE REG
0BB4 34 08 0C5A 5222 ST IUS175+@OP1,@ARR SET RETURN BRANCH ADDRESS 1-3
5223 *
5224 * ACCESS VIRTUAL PAGE TO CONTAIN ELEMENT FIRST (OR ONLY) SEGMENT
5225 *
0BB8 C0 87 1349 5226 IUS010 B IPGMOD LINK TO GET PAGE, CONVERT THE
5227 * * VADDR, AND SET PAGE MODIFY
5228 *
5229 * TEST WHETHER STACKED DATA TYPE IS TO BE MATCHED WITH VM DATA TYPE -
5230 * MATCHING IS REQUIRED ONLY FOR ASSIGNMENTS FROM DATA FILES.
5231 *
0BBC 35 01 144C 5232 IUS012 L IZCADR,@BR LOAD THE 1ST SEGMENT CADDR
5233 *
0BC0 F2 00 1B 5234 IUS014 JC IUS025,*-* GO TEST TRACE INDICATOR IF
0BC1 5235 ORG IUS014+@Q * ELEMENT TYPES NEED NOT BE
0BC1 87 0BC1 5236 DC AL1(@UCB) * MATCHED - INITIALIZE SWITCH
0BC3 5237 ORG IUS014+@INST3 * TO BYPASS ELEMENT TYPE MATCH
5238 *
5239 * DATA MATCHING REQUIRED - COMPARE DATA ELEMENT TYPE INDICATORS
5240 *
0BC3 3C 90 0BD5 5241 IUS016 MVI IUS020+@Q,@BF SET STACKED DATA TEST FOR ARITH
0BC7 78 40 00 5242 TBN I@STAT(,@BR),B@DTYP IF DESTINATION VARIABLE = ARITH
0BCA F2 90 04 5243 JF IUS018 * SKIP TO TEST FOR A TYPE MATCH
0BCD 3C 10 0BD5 5244 MVI IUS020+@Q,@BT SET STACKED DATA TEST FOR CHAR
0BD1 B8 40 00 5245 IUS018 TBN I@STAT(,@XR),B@DTYP TEST V.M. ELEMENT TYPE INDICATOR
0BD4 F2 00 07 5246 IUS020 JC IUS025,*-* BRANCH IF DATA TYPES ARE EQUAL
5247 *
5248 * DATA MISMATCH - SET 'INVALID VARIABLE ASSIGNMENT' ERROR MESSAGE
5249 *
0BD7 3C C4 0CBC 5250 IUS022 MVI IZERRC,@@E727 SET INTERPRETER ERROR CODE
0BDB F2 87 CD 5251 J INTERR GO TERMINATE ON DATA ERROR 1-3
5252 *
5253 *****

```

## S/3 BASIC INTERPRETER ELEMENT STACKING RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 53
		5255		*****	
		5256	*	TEST TRACE INDICATOR IN PROGRAM VARIABLE BEING MODIFIED	
		5257		*****	
		5258	*		
0BDE 78 80 00		5259	IUS025 TBN	I@STAT(,@BR),B@TRAC	TEST VARIABLE TRACE INDICATOR
0BE1 C2 01 0BB8		5260	LA	IUS010,@BR	LOAD IUSTAK BASE REGISTER
0BE5 BB 80 00		5261	SBF	I@STAT(,@XR),B@TRAC	VARIABLE TRACE INDICATOR OFF 1-3
0BE8 F2 90 0C		5262	JF	IUS040	BRANCH IF TRACE INDICATOR OFF
		5263	*		
		5264	*	TRACE INDICATOR ON - SET CONDITIONS TO DISPLAY THE VARIABLE	
		5265	*		
0BEB BA 80 00		5266	IUS030 SBN	I@STAT(,@XR),B@TRAC	RETAIN VARIABLE TRACE INDICATOR
0BEE 0C 01 0D57 144A		5267	MVC	IZPARM,IZVADR(@VADDR)	SAVE THE VARIABLE VIRTUAL ADDR
0BF4 7C 80 8C		5268	MVI	IUS150+@Q(,@BR),@NOP	ENABLE VARIABLE DISPLAY ROUTINE
		5269	*		
		5270		*****	
		5272		*****	
		5273	*	COMPLETE THE FINAL STACK/VM MOVE INSTRUCTION - THE 2ND DISPLACEMENT	
		5274	*	IN THIS INSTRUCT/ON CONTAINS THE ELEMENT LENGTN CODE INPUT PARAMETER	
		5275		*****	
		5276	*		
0BF7 5C 01 81 82		5277	IUS040 MVC	IUS120+@DD2-1(,@BR),IUS120+@DD2(@INST4-2,@BR)	PROPAGATE
		5278	*		* THE ELEMENT LENGTH PARAMETER
		5279	*		
		5280	*	TEST FOR POSSIBLE VIRTUAL PAGE BOUNDARY CONDITION	
		5281	*		
0BFB 4C 00 68 144A		5282	IUS050 MVC	IUS090+@Q(,@BR),IZPGDS(@VADDR-1)	COMPUTE 2ND SEGMENT
0C00 5E 00 68 82		5283	ALC	IUS090+@Q(,@BR),IUS120+@DD2(1,@BR)	* LENGTH PARAMETER -
		5284	*		* IF ELEMENT WILL IN ONE
0C04 F2 82 2C		5285	JL	IUS110	* PAGE. GO UNSTACK AS A UNIT
		5286	*		
		5287		*****	

## S/3 BASIC INTERPRETER ELEMENT STACKING RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 54
		5289		*****	
		5290	*	PAGE BOUNDARY CONDITION EXISTS - ESTABLISH THE ELEMENT FIRST	
		5291	*	SEGMENT UNSTACKING INSTRUCTION	
		5292		*****	
		5293	*		
0C07	7C FF 62	5294	IUS060 MVI	IUS080+@DD2(,@BR),B@LVPG-1 CALCULATE 1ST SEGMENT	
0C0A	4F 00 62 144A	5295	SLC	IUS080+@DD2(,@BR),IZPGDS(@VADDR-1) * LENGTH PARAMETER	
0C0F	5C 01 61 62	5296	MVC	IUS080+@DD2-1(,@BR),IUS080+@DD2(@INST4-2,@BR) PROPAGATE	
		5297	*	* IF 1ST SEGMENT LENGTH PARM	
		5298	*		
		5299	*	UNSTACK THE 1ST OF 2 ELEMENT SEGMENTS	
		5300	*		
0C13	35 01 144C	5301	IUS070 L	IZCADR,@BR LOAD THE FIRST SEGMENT CADDR	
0C17	6C 00 00 00	5302	IUS080 MVC	*-*(,@BR),*-*(@VQ,@XR) MOVE 1ST SEGMENT TO CORE PAGE	
0C1B	C2 01 0BB8	5303	LA	IUS010,@BR RESTORE IUSTAK BASE ADDRESS	
		5304	*		
		5305	*	ESTABLISH CONDITIONS TO UNSTARK THE ELEMENT SECOND SEGMENT	
		5306	*		
0C1F	7C 00 81	5307	IUS090 MVI	IUS120+@D1(,@BR),*-* SET UNSTACKING INST	
0C22	5C 00 80 81	5308	MVC	IUS120+@Q(,@BR),IUS120+@D1(1,@BR) * DISP & LENGTH FIELDS	
		5309	*		
0C26	1E 00 1449 A3	5310	ALC	IZPGNO,IUSBN1(@VADDR-1,@BR) ADJUST PAGING ROUTINE PARM	
0C2B	3C 00 144A	5311	MVI	IZPGDS,@ZERO * TO REFERENCE NEXT V.M. PAGE	
		5312	*		
		5313	*	ACCESS VIRTUAL PAGE TO CONTAIN ELEMENT SECOND SEGMENT	
		5314	*		
0C2F	C0 87 1349	5315	IUS100 B	IPGMOD LINK TO GET PAGE, CONVERT THE	
		5316	*	* VADDR, AND SET PAGE MODIFY	
		5317	*		
		5318		*****	

## S/3 BASIC INTERPRETER ELEMENT STACKING RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 55
		5320		*****	
		5321		* UNSTACK THE ELEMENT FINAL (SECOND OR ONLY) SEGMENT	
		5322		*****	
		5323		*	
0C33	35 01 144C	5324	IUS110 L	IZCADR,@BR	LOAD TNE FINAL SEGMENT CADDR
		5325		*	
0C37	6C 00 00 00	5326	IUS120 MVC	*-*(,@BR),*-*(@VQ,@XR)	MOVE FINAL SEGMENT TO CORE PAGE
0C3A		5327		ORG IUS120+@DD2	INITIALIZE TNE FINAL MOVE INST
0C3A	04	0C3A 5328		DC AL1(I@LPFV-1)	* TO UNSTACK CURRENT PRECISION
0C3B		5329		ORG IUS120+@INST4	* ARITHMETIC ELEMENT
		5330		*	
0C3B	3C 04 0C3A	5331	IUS130 MVI	IUS120+@DD2,I@LPFV-1	RESET THE ELEMENT LENGTN CODE
		5332		*	* INPUT PARM FOR ARITH ELEMENT
		5333		*	
		5334		* EXIT - RETURN CONTROL TO THE CALLING PROGRAM	
		5335		*	
0C3F	C2 01 0000	5336	IUS140 LA	*-*,@BR	RESTORE CALLING PROGRAM BASE
		5337		*	
0C43	F2 00 11	5338	IUS150 JC	IUS175,*-*	RETURN TO CALLING PROG IF 1-3
0C44		5339		ORG IUS150+@Q	* VARIABLE TRACE IS DISABLED -
0C44	87	0C44 5340		DC AL1(@UCB)	* INITIALIZE BRANCH CONDITION
0C46		5341		ORG IUS150+@INST3	* TO DISABLE THE TRACE SW. 1-3
		5342		*	
		5343		* VARIABLE TRACE MODE - TEST FOR ENABLED TRACE CONDITION	
		5344		*	
0C46	3C 87 0C44	5345	IUS160 MVI	IUS150+@Q,@UCB	DISABLE VARIABLE TRACE ROUTINE
0C4A	38 04 03D0	5346		TBN \$XIND1,\$TRACE	IF EXECUTION TRACE NOT ENABLED
0C4E	F2 90 06	5347		JF IUS175	* GO EXIT TO CALLING PROGRAM 1-3
		5348		*	
		5349		* TRACE ENABLED - DISPLAY MODIFIED VARIABLE AND EXIT	
		5350		*	
0C51	C0 87 130B	5351	IUS170 B	IPGCAL	LINK TO DISPLAY VARIABLE VALUE
0C55	4700	0C56 5352		DC AL(@VADDR)(V\$DTRV)	VARIABLE TRACE RTN ENTRY VADDR
		5353		*	
0C57	C0 87 0000	5354	IUS175 B	*-*	GO EXIT TO CALLING PROGRAM 1-3
		5355		*	
		5356		*****	

## S/3 BASIC INTERPRETER ELEMENT STACKING RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 56
		5358		*****	
		5359		* ELEMENT UNSTACKING ROUTINE CONSTANTS	
		5360		*****	
		5361		*	
0C5B 01		0C5B	5362	IUSBN1 DC IL1'1' BINARY INTEGER +1	
		5363		*	
		5364		*****	
		5365		* ELEMENT UNSTACKING ROUTINE EQUATES REFERENCING PROGRAM	
		5366		*****	
		5367		*	
		0C3A	5368	IUSLNG EQU IUS120+@DD2 ELEMENT LENGTH CODE INPUT PARAM	
			5369	* (ELEMENT LENGTH - 1)	
			5370	*	
		0BC1	5371	IUSDSW EQU IUS014+@Q DATA TYPE MATCHING SWITCH	
			5372	* @NOP ENABLE DATA TYPE MATCH	
			5373	* @UCB DISABLE DATA TYPE MATCH	
			5374	*	
			5375	* END OF ELEMENT UNSTACKING ROUTINE CODING	
			5376	*	
			5377	*****	
			5378	* INTERPRETER EXECUTIVE ROUTINE INITIAL ENTRT POINT	
			5379	*****	
			5380	*	
			5381	* ENTER INTERP - ESTABLISH ADDRESSABILITY	
			5382	*	





## S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 58
		5385		*****	*
		5386	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		5387	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
		5388	*		*
		5389		*****	*
		5390	*	*STATUS	*
		5391	*	VERSION 1 MODIFICATION 0	*
		5392	*		*
		5393	*	*FUNCTION	*
		5394	*	* INTERP HAS THE PRIMARY FUNCTION OF TRANSLATING A PSEUDO INSTRUCTIONS	*
		5395	*	OPCODE TO AN EXECUTION ROUTINE CORE ADDRESS, THEN PASSING	*
		5396	*	CONTROL TO THAT PMC PROCESSOR. IT ALSO CONTAINS HOUSEKEEPING	*
		5397	*	ROUTINES AND WORK AREAS WHICH ARE CENTRAL TO INTERPRETER OPERATIONS,	*
		5398	*	IN ADDITION TO THE RUN-TIME ROUTINES WHICH EXECUTE THE	*
		5399	*	FOLLOWING PSEUDO MACHINE INSTRUCTIONS -	*
		5400	*	* 'STH' - STATEMENT HEADER	*
		5401	*	* 'IMH' - IMAGE STATEMENT HEADER	*
		5402	*	* 'HLT' - HALT EXECUTION	*
		5403	*	* 'EOP' - END OF PMC PAGE	*
		5404	*	* 'SVC' - SUPERVISOR CALL	*
		5405	*	* INTERP PRIMARY FUNCTIONS ARE BEST DESCRIBED IN TERMS OF THE	*
		5406	*	VARIOUS ENTRY POINTS TO THE MODULE. THESE DESCRIPTIONS APPEAR	*
		5407	*	BELOW UNDER 'ENTRY POINTS'.	*
		5408	*	* THE FOLLOWING DESCRIPTIONS GIVE FUNCTIONAL SPECIFICATIONS FOR	*
		5409	*	THOSE ROUTINES (INTERNAL TO INTERS)) WHICH ARE USED TO EXECUTE	*
		5410	*	THE PSEUDO MACHINE INSTRUCTIONS LISTED ABOVE.	*
		5411	*	* 'STH' - STATEMENT HEADER (FORMAT - OP LINE)	*
		5412	*	THE PSEUDO INSTRUCTION SEQUENCE FOR EACH TRANSLATED BASIC	*
		5413	*	STATEMENT (EXCEPT 'IMAGE') BEGINS WITH AN 'STH' INSTRUCTION.	*
		5414	*	SYSTEM STATUS IS TESTED, AND EXECUTION IS HALTED IN	*
		5415	*	THE 'PAUSE' STATE IF AN INTERRUPT CONDITION OR 'STEP' MODE	*
		5416	*	OPERATION IS IN EFFECT (SEE 'HLT'). OTHERWISE, BINARY	*
		5417	*	LINE NUMBER 'LINE' IS SAVED FOR GENERAL REFERENCE AND CONTROL	*
		5418	*	IS PASSED TO THE NEXT SEQUENTIAL INSTRUCTION.	*
		5419	*	* 'IMH' - IMAGE STATEMENT HEADER (FORMAT - OP LINE)	*
		5420	*	THE PSEUDO INSTRUCTION FOR EACH TRANSLATED BASIC 'IMAGE'	*
		5421	*	STATEMENT BEGINS WITH AN 'IMH' INSTRUCTION. WHEN EXECUTED	*
		5422	*	AS THE OBJECT OF A 'PRINT USING' CALL, 'IMH' ACTS AS A	*
		5423	*	NO-OP. OTHERWISE, 'IMH' EXECUTION IS IDENTICAL TO 'STH'	*
		5424	*	EXECUTION.	*
		5425	*	* 'HLT' - HALT EXECUTION (FORMAT - OP)	*
		5426	*	PMC EXECUTION IS HALTED IN THE 'PAUSE' STATE, AND PROGRAM	*
		5427	*	STATUS IS PRESERVED SO THAT EXECUTION CAN BE RESUMED OR	*
		5428	*	ABORTED AS DESIRED UNDER SYSTEM CONTROL.	*
		5429	*	* 'EOP' - END OF PMC PAGE (FORMAT - OP)	*
		5430	*	EACH PMC VIRTUAL PAGE IS TERMINATED WITH AT LEAST ONE	*
		5431	*	'EOP' INSTRUCTION. 'EOP' EXECUTION RESULTS IN CONTROL	*
		5432	*	BEING PASSED TO THE FIRST PSEUDO INSTRUCTION APPEARING IN	*
		5433	*	THE NEXT SEQUENTIAL VIRTUAL PAGE.	*
		5434	*	* 'SVC' - SUPERVISOR CALL (FORMAT - OP)	*
		5435	*	CONTROL IS PASSED TO THE SYSTEM SUPERVISOR AT THE END OF	*
		5436	*	PROGRAM EXECUTION. ALL ACTIVE EXTERNAL DATA FILES ARE	*
		5437	*	CLOSED, CORE-RESIDENT VIRTUAL MEMORY PAGES ARE RESTORED TO	*
		5438	*	DISK, AND PROGRAM EXECUTION CANNOT BE RESUMED.	*
		5439	*	IF IN THE LINE PRINTER MODE, ANY REMAINING DATA TO BE	*
		5440	*	PRINTED WILL BE PRINTED.	*

## S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 59
		5441	*		*
		5442	*	ENTRY POINTS	*
		5443	*	* ENTRY INTERP - THE FIRST VIRTUAL MEMORY PMC PAGE IS LOCKED INTO	*
		5444	*	CORE. THE FIRST PSEUDO INSTRUCTION IN THE PAGE IS ACCESSED,	*
		5445	*	AND CONTROL IS PASSED TO THE APPROPRIATE OMC EXECUTION ROUTINE.	*
		5446	*	CALLING SEQUENCE IS	*
		5447	*	B INTERP	*
		5448	*	SUBJECT TO INPUT CONDITIONS DESCRIBED BELOW.	*
		5449	*	* ENTRY INTRAG - THE VIRTUAL MEMORY PMC PAGE SPECIFIED IN PAGING	*
		5450	*	PARAMETER I2VADR IS LOCKED INTO CORE. THE PSEUDO INSTRUCTION	*
		5451	*	REFERENCED BY I2VADR IS ACCESSED, AND CONTROL IS PASSED TO THE	*
		5452	*	APPROPRIATE PMC EXECUTION ROUTINE. CALLING SEQUENCE IS	*
		5453	*	B INTPAG	*
		5454	*	SUBJECT TO INPUT CONDITIONS DESCRIBED BELOW. ENTRY POINT	*
		5455	*	INTPAG MAY ALSO BE SPECIFIED AS I\$NPAG WHEN REFERENCED IN ONE	*
		5456	*	OR THE SUBROUTINES IN VIRTUAL MEMORY.	*
		5457	*	* ENTRY INTXEC - THE PSEUDO MACHINE INSTRUCTION REFERENCED BY	*
		5458	*	PMC ADDRESS REGISTER INTIAR IS ACCESSED, AND CONTROL IS PASSED	*
		5459	*	TO THE APPROPRIATE PMC EXECUTION ROUTINE. CALLING SEQUENCE IS	*
		5460	*	B INTXEC	*
		5461	*	SUBJECT TO INPUT CONDITIONS DESCRIBED BELOW.	*
		5462	*	* ENTRY INTAD1 - THE PSEUDO INSTRUCTION ADDRESS REGISTER (INTIAR)	*
		5463	*	IS INCREMENTED PAST A 1-BYTE INSTRUCTION. THE PSEUDO INSTRUC-	*
		5464	*	TION NOW REFERENCED BY INTIAR IS ACCESSED, AND CONTROL IS	*
		5465	*	PASSED TO THE APPROPRIATE PMC EXECUTION ROUTINE. CALLING	*
		5466	*	SEQUENCE IS	*
		5467	*	B INTAD1	*
		5468	*	SUBJECT TO INPUT CONDITIONS SPECIFIED BELOW. ENTRY POINT	*
		5469	*	INTAD1 MAY ALSO BE SPECIFIED AS I\$XAD1 WHEN REFERENCED IN ONE	*
		5470	*	OF THE SUBROUTINES IN VIRTUAL MEMORY.	*
		5471	*	* ENTRY INTAD2 - THE PSEUDO INSTRUCTION ADDRESS REGISTER (INTIAR)	*
		5472	*	IS INCREMENTED PAST A 2-BYTE INSTRUCTION. THE PSEUDO INSTRUC-	*
		5473	*	TION NOW REFERENCED BY INTIAR IS ACCESSED, AND CONTROL IS	*
		5474	*	PASSED TO THE APPROPRIATE PMC EXECUTION ROUTINE. CALLING	*
		5475	*	SEQUENCE IS	*
		5476	*	B INTAD2	*
		5477	*	SUBJECT TO INPUT CONDITIONS SPECIFIED BELOW. ENTRY POINT	*
		5478	*	INTAD2 MAY ALSO BE SPECIFIED AS I\$XAD2 WHEN REFERENCED IN ONE	*
		5479	*	OF THE SUBROUTINES IN VIRTUAL MEMORY.	*
		5480	*	* ENTRY INTAD3 - THE PSEUDO INSTRUCTION ADDRESS REGISTER (INTIAR)	*
		5481	*	IS INCREMENTED PAST A 3-BYTE INSTRUCTION. THE PSEUDO INSTRUC-	*
		5482	*	TION NOW REFERENCED BY INTIAR IS ACCESSED, AND CONTROL IS	*
		5483	*	PASSED TO THE APPROPRIATE PMC EXECUTION ROUTINE. CALLING	*
		5484	*	SEQUENCE IS	*
		5485	*	B INTAD3	*
		5486	*	SUBJECT TO INPUT CONDITIONS SPECIFIED BELOW. ENTRY POINT	*
		5487	*	INTAD3 MAY ALSO BE SPECIFIED AS I\$XAD3 WHEN REFERENCED IN ONE	*
		5488	*	OF THE SUBROUTINES IN VIRTUAL MEMORY.	*
		5489	*	* ENTRY INTAD4 - THE PSEUDO INSTRUCTION ADDRESS REGISTER (INTIAR)	*
		5490	*	IS INCREMENTED PAST A 4-BYTE INSTRUCTION. THE PSEUDO INSTRUC-	*
		5491	*	TION NOW REFERENCED BY INTIAR IS ACCESSED, AND CONTROL IS	*
		5492	*	PASSED TO THE APPROPRIATE PMC EXECUTION ROUTINE. CALLING	*
		5493	*	SEQUENCE IS	*
		5494	*	B INTAD4	*
		5495	*	SUBJECT TO INPUT CONDITIONS SPECIFIED BELOW. ENTRY POINT	*
		5496	*	INTAD4 MAY ALSO BE SPECIFIED AS I\$XAD4 WHEN REFERENCED IN ONE	*

## S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 06/09/20 PAGE 60

```

5497 *      OF THE SUBROUTINES IN VIRTUAL MEMORY. *
5498 *      * ENTRY INTADS - THE RUN-TIME STACK POINTER (INTSTP) IS INCREMEN- *
5499 *      TED BY THE VALUE IN PARAMETER INTST1.  AN ERROR CONDITION *
5500 *      OCCURS WHEN INTSTP IS INCREMENTED BEYOND THE STACK DATA LIMIT, *
5501 *      CALLING SEQUENCE IS *
5502 *      B      INTADS *
5503 *      SUBJECT TO INPUT CONDITIONS SPECIFIED BELOW.  ENTRY POINT *
5504 *      INTADS MAY ALSO BE SPECIFIED AS I$ADST WHEN REFERENCED IN ONE *
5505 *      OF THE SUBROUTINES IN VIRTUAL MEMORY. *
5506 *      * ENTRY INTERR - THE CODE IN ERROR PARAMETER INTERC IS STORED AS *
5507 *      A PARAMETER TO THE SYSTEM ERROR PROGRAM.  ALL ACTIVE EXTERNAL *
5508 *      DATA FILES ARE CLOSED.  CORE-RESIDENT VIRTUAL MEMORY PAGES ARE *
5509 *      RESTORED TO DISK, AND CONTROL IS PASSED TO THE ERROR PROGRAM TO *
5510 *      TERMINATE EXECUTION.  CALLING SEQUENCE IS *
5511 *      B      INTERR *
5512 *      SUBJECT TO INPUT CONDITIONS SPECIFIED BELOW.  ENTRY POINT *
5513 *      INTERR MAY ALSO BE SPECIFIED AS I$XERR WHEN REFERENCED IN ONE *
5514 *      OF THE SUBROUTINES IN VIRTUAL MEMORY. *
5515 *      * ENTRY INT700 - THE LINE PRINTER BUFFER IS CHECKED IF EMPTY. *
5516 *      IF NOT, THE DATA WILL BE PRINTED VIA A CALL TO DLPRT. *
5517 *      THE CALLING SEQUENCE IS *
5518 *      B      I$1700 *
5519 * * * * * *
5520 * INPUT *
5521 *      * INTXPG (EXTERNAL IZXPAG, I$YPAG - FOR ENTRY POINT INTPAG) - *
5522 *      1 BYTE, FOR THE EXECUTION PAGE NUMBER.  THIS CONTAINS THE *
5523 *      VIRTUAL PAGE NUMBER OF THE PMC PAGE TO WHICH CONTROL IS TO BE *
5524 *      TRANSFERRED. *
5525 *      * IZVADR (FOR ENTRY POINT INTPAG) - 2 BYTES, FOR THE PAGING *
5526 *      ROUTINE VIRTUAL ADDRESS PARAMETER.  THIS CONTAINS THE VIRTUAL *
5527 *      ADDRESS OF THE PSEUDO INSTRUCTION TO WHICH CONTROL IS TO BE *
5528 *      TRANSFERRED. *
5529 *      * INTIAR (EXTERNAL IZXIAR, I$XTAR - FOR ENTRY POINT INTXEC) - *
5530 *      2 BYTES, FOR THE PMC ADDRESS REGISTER.  THIS CONTAINS THE CORE *
5531 *      ADDRESS OF THE OPCODE BYTE IN THE PSEUDO INSTRUCTION TO BE *
5532 *      EXECUTED. *
5533 *      * INTIAR (EXTERNAL IZXIAR, I4XIAR) - FOR ENTRY POINTS INTAD1 TILL *
5534 *      INTAD4) - 2 BYTES, FOR THE PMC ADDRESS REGISTER.  THIS CONTAINS *
5535 *      THE CORE ADDRESS OF THE OPCODE BYTE IN THE PSEUDO INSTRUCTION *
5536 *      TO BE INCREMENTED PAST. *
5537 *      * INTSTI (EXTERNAL IZSTKI, I$STKI - FOR ENTRY POINT INTADS) - *
5538 *      1 BYTE, FOR THE RUN-TIME STACK POINTER INCREMENT.  THIS CON- *
5539 *      TAINS THE VALUE OF THE INCREMENT TO BE ADDED TO INTSTP. *
5540 *      * INTERC (EXTERNAL IZERRC, I$ERRC - FOR ENTRY POINT INTERR) - *
5541 *      1 BYTE, FOR THE INTERPRETER ERROR CODE.  THIS CONTAINS THE CODE *
5542 *      ASSOCIATED WITH THE ERROR MESSAGE TO BE DISPLAYED BY THE SYSTEM *
5543 *      ERROR PROGRAM ON EXIT TO $CAERK. *
5544 *      * INTISW (EXTERNAL IZIRSW, I$IRSW - FOR 'STH'/'IMH' EXECUTION) - *
5545 *      1 BYTE, FOR THE IMAGE REFERENCE SWITCH.  THIS SWITCH, NORMALLY *
5546 *      SET TO CODE @NOP (OFF), IS SET TO CODE $UCB (ON) WHEN THE *
5547 *      STATEMENT HEADER TO BE EXECUTED MUST BE AN 'IMH' INSTRUCTION *
5548 *      RATHER THAN AN 'STH'.  'STH' EXECUTION WITH THIS SWITCH SET ON *
5549 *      CAUSES ERROR 3 (SEE ERROR PROCEDURES). *
5550 *      * INTRSW (EXTERNAL IZRESW, I$RESW - FOR 'STH' EXECUTION) - 1 BYTE, *
5551 *      FOR THE STATEMENT RECURSION ERROR SWITCH.  THIS IS SET TO CODE *
5552 *      @NOP (OFF) WHEN LINE NUMBER RECURSION IS PERMITTED DURING 'STH' *

```

## S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 61
		5553	*	EXECUTION. UNLESS SPECIFICALLY SET PRIOR TO EACH 'STN' INSTRU-	*
		5554	*	TION EXECUTION, INTRSW CONTAINS CODE @UCB (ON) WHICH CAUSES AN	*
		5555	*	ERROR CONDITION WHEN LINE NUMBER RECURSION OCCURS.	*
		5556	*	* INTTSW (EXTERNAL IZTFSW, ISTFSW - FOR 'STH' EXECUTION) - 1 BYTE,	*
		5557	*	FOR THE TRACE FLOW SWITCH. THIS IS SET TO CODE @NOP (ON) WHEN	*
		5558	*	'TRACE FLOW' IS SPECIFIED, AND CAUSES LINE NUMBER DISPLAY WHEN	*
		5559	*	INDICATOR BIT \$TRACE IN \$XIND1 IS ALSO ON. WHEN 'TRACE' MODE	*
		5560	*	PROCESSING HAS NOT BEEN SPECIFIED AT THE START OF EXECUTION,	*
		5561	*	INTTSW IS SET TO CODE @UCB (OFF).	*
		5562	*	* \$INLNO (FOR 'STH' EXECUTION) - 2 BYTES, FOR THE SYSTEM EXECU-	*
		5563	*	TION LINE NUMBER. THIS CONTAINS THE BINARY LINE NUMBER OR THE	*
		5564	*	LAST EXECUTED STATEMENT, OR THE VALUE X'FFFF' WHEN THE FIRST	*
		5565	*	'STH' INSTRUCTION IN THE PROGRAM IS TO BE EXECUTED.	*
		5566	*		*
		5567	*	*OUTPUT	*
		5568	*	* INTIAR (EXTERNAL IZXIAR, I\$XIAR - AFTER ENTRY POINT INTPAG) -	*
		5569	*	2 BYTES, FOR THE PMC ADDRESS REGISTER. THIS CONTAINS THE CORE	*
		5570	*	ADDRESS OF THE OPCODE BYTE IN THE PSEUDO INSTRUCTION TO WHICH	*
		5571	*	CONTROL IS TRANSFERRED.	*
		5572	*	* INTIAR (EXTERNAL IZXIAR, I\$XIAR - AFTER ENTRY POINTS INTAD1,	*
		5573	*	INTAD2, INTAD3, INTAD4) - 2 BYTES, FOR THE PMC ADDR REGISTER.	*
		5574	*	THIS CONTAINS THE CORE ADDRESS OF THE OPCODE BYTE IN THE PSEUDO	*
		5575	*	INSTRUCTION FOLLOWING THAT WHICH HAS BEEN INCREMENTED PAST.	*
		5576	*	* INTSTP (EXTERNAL IZSTAK, I\$STAK - AFTER ENTRY POINT INTADS) -	*
		5577	*	2 BYTES, FOR THE RUN-TIME STACK POINTER. THIS HAS BEEN INCRE-	*
		5578	*	MENTED BY THE VALUE IN PARAMETER INTSTI.	*
		5579	*	* \$CAERR (AFTER ENTRY POINT INTERR) - 1 BYTE, FOR THE SYSTEM	*
		5580	*	ERROR PROGRAM MESSAGE CODE PARAMETER. THIS IS SET EQUAL TO THE	*
		5581	*	CODE STORED IN INTERC.	*
		5582	*	* \$INLNO (AFTER 'STH' EXECUTION) - 2 BYTES, FOR THE SYSTEM EXECU-	*
		5583	*	TION LINE NUMBER. THIS IS SET TO CONTAIN THE BINARY LINE NUM-	*
		5584	*	BER OPERAND IN THE 'STH' INSTRUCTION.	*
		5585	*	INTSHA (EXTERNAL IZSTHA, I\$STHA - AFTER 'STH' EXECUTION) -	*
		5586	*	2 BYTES, FOR THE STATEMENT HEADER VIRTUAL ADDRESS. THIS IS SET	*
		5587	*	TO CONTAIN THE VIRTUAL ADDRESS OF THE OPCODE IN THE CURRENTLY	*
		5588	*	EXECUTED 'STH' INSTRUCTION.	*
		5589	*	* INTRSW (EXTERNAL IZRESW, I\$RESW - AFTER 'STH' EXECUTION) -	*
		5590	*	1 BYTE, FOR THE STATEMENT RECURSION ERROR SWITCH. 'STH' EXECU-	*
		5591	*	TION ALWAYS CAUSES THIS SWITCH TO BE RESET TO THE ON CONDITION	*
		5592	*	(CODE @UCB).	*
		5593	*	* INTISW (EXTERNAL IZIRSW, I\$IRSW - AFTER 'IMH' EXECUTION) -	*
		5594	*	1 BYTE, FOR THE IMAGE REFERENCE SWITCH. THIS SWITCH IS SET OFF	*
		5595	*	(CODE @NOP) DURING 'IMH' INSTRUCTION EXECUTION.	*
		5596	*	* \$XIND2 (AFTER ENTRY POINT INTERR OR 'SVC' EXECUTION) - 1 BYTE,	*
		5597	*	FOR SYSTEM EXECUTION INDICATOR 2. BIT \$EXCMD IS SET OFF, INDI-	*
		5598	*	CATING TERMINATION OF EXECUTION MODE.	*
		5599	*	* EXTERNAL DATA FILES (AFTER ENTRY POINT INTERR OR 'SVC' EXECU-	*
		5600	*	TION) - ALL ACTIVE EXTERNAL DATA FILES ARE CLOSED BEFORE EXECU-	*
		5601	*	TION IS TERMINATED.	*
		5602	*	* DISK VIRTUAL MEMORY (AFTER ENTRY POINT INTERR OR 'SVC' EXECU-	*
		5603	*	TION) - ALL MODIFIED CORE PAGES ARE WRITTEN BACK TO DISK VIR-	*
		5604	*	TUAL MEMORY BEFORE EXECUTION IS TERMINATED,	*
		5605	*	* PMC EXECUTION CONTROL (AFTER ENTRY POINTS INTPAG, INTAD1 INTAD2,	*
		5606	*	INTAD3, INTAD4, INTXEC) - CONTROL IS PASSED TO THE CORE-RESIDENT	*
		5607	*	* PMC EXECUTION ROUTINE DEFINED BY THE INSTRUCTION OPCODE REFER-	*
		5608	*	ENCED BY THE OUTPUT ADDRESS IN INTIAR.	*



## S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 62
		5609	*		*
		5610	*	*EXTERNAL REFERENCES	*
		5611	*	* \$CARPL - NUCLEUS ENTRY POINT TO LOAD AND EXECUTE #GUFUD.	*
		5612	*	* \$CAERK - NUCLEUS ENTRY POINT TO LOAD AND EXECUTE #ERRPG.	*
		5613	*	* \$PAUSD - NUCLEUS ENTRY POINT TO LOAD AND EXECUTE #EXMSG.	*
		5614	*	* \$UNMSK - NUCLEUS ENTRY POINT TO TEST FOR CONSOLE INTERRUPT.	*
		5615	*	* IPGCAL - ENTRY POINT FOR PAGING MODULE V.M. PROGRAM CALL RTN.	*
		5616	*	* IPGLOK - ENTRY POINT FOR PAGING MODULE V.M. PAGE LOCKING RTN.	*
		5617	*	* IPGULK - ENTRY POINT FOR PAGING MODULE V.M. PAGE UNLOCKING RTN.	*
		5618	*	* V\$DTLN - VIRTUAL ENTRY ADDRESS FOR FZLINT, LINE NO. TRACE RTN.	*
		5619	*	* V\$VMPS - VIRTUAL ENTRY ADDRESS FOR FZZVPS, V.M. PUSW ROUTINE.	*
		5620	*	* V\$XKCA - VIRTUAL ENTRY ADDRESS FOR SFRCAL, CLOSE ALL FILES RTN.	*
		5621	*	* V\$LPRT - VIRTUAL ENTRY ADDRESS FOR DLFprt, LINE PRINTER RTN.	*
		5622	*	* \$CAERR - 1 BYTE, FOR THE ERROR CODE PARAMETER TO #ERRPG.	*
		5623	*	* \$CIMSK - 1 BYTE, FOR THE CONSOLE INTERRUPT ENABLE MASK.	*
		5624	*	* \$INLNO - 2 BYTES, FOR THE CURRENT EXECUTION LINE NUMBER.	*
		5625	*	* \$XIND1 - 1 BYTE, FOR SYSTEM EXECUTION INDICATOR 1.	*
		5626	*	* \$XIND2 - 1 BYTE, FOR SYSTEM EXECUTION INDICATOR 2.	*
		5627	*	* IZPGNO - 1 BYTE, FOR PAGING MODULE VIRTUAL PAGE NO. PARAMETER.	*
		5628	*	* IZPGDS - 1 BYTE, FOR PAGING MODULE VIRTUAL PAGE DISP PARAMETER.	*
		5629	*	* IZCADS - 2 BYTES, FOR PAGING MODULE CORE ADDRESS OUTPUT PARAM.	*
		5630	*	* IZSTKB - CORE ADDRESS OF RUN-TIME STACK LEFTMOST BYTE.	*
		5631	*	* IN ADDITION TO THOSE ETERNAL REFERENCES SPECIFIED ABOVE, THE	*
		5632	*	INTERPRETER PMC EXECUTION BRANCH ADDRESS TABLE (INTBAT)	*
		5633	*	CONTAINS ENTRY POINT ADDRESSES FOR EACH CORE-RESIDENT ROUTINE	*
		5634	*	REQUIRED TO PROCESS GENERATED PSEUDO MACHINE INSTRUCTIONS.	*
		5635	*		*
		5636	*	*EXITS, NORMAL	*
		5637	*	* ENTRY POINT INTADS - CONTROL IS NORMALLY RETURNED TO THE FIRST	*
		5638	*	INSTRUCTION FOLLOWING THE CALLING SEQUENCE.	*
		5639	*	* ENTRY POINT INTERR - CONTROL IS TRANSFERRED TO SYSTEM ERROR	*
		5640	*	PROGRAM #ERRPG THROUGH ENTRY ADDRESS \$CAERK. CONDITIONS ESTAB-	*
		5641	*	LISHED WHEN THIS EXIT IS TAKEN ARE SPECIFIED AS 'OUTPUT' ABOVE.	*
		5642	*	* ALL OTHER ENTRY POINTS - CONTROL IS PASSED TO THE CORE-RESIDENT	*
		5643	*	PMC EXECUTION ROUTINE DEFINED BY THE OPCODE IN THE REFERENCED	*
		5644	*	PSEUDO INSTRUCTION.	*
		5645	*	* 'SVC' EXECUTION - CONTROL IS TRANSFERRED TO SYSTEM FILE UPDATER	*
		5646	*	#GUFUD THROUGH ENTRY ADDRESS \$CARPL. CONDITIONS ESTABLISHED	*
		5647	*	WHEN THIS EXIT IS TAKEN ARE SPECIFIED AS 'OUTPUT' ABOVE.	*
		5648	*		*
		5649	*	*EXITS, ERROR	*
		5650	*	* ENTRY POINT INTADS - CONTROL IS PASSED TO ERROR ROUTINE INTERR	*
		5651	*	WHEN THE RUN-TIME STACK IS FILLED BEYOND CAPACITY (SEE ERROR	*
		5652	*	PROCEDURES BELOW).	*
		5653	*	* 'STH' EXECUTION - CONTROL IS PASSED TO ERROR ROUTINE INTERR	*
		5654	*	WHEN STATEMENT LINE NUMBER RECURSION OCCURS UNDER ADVERSE CON-	*
		5655	*	DITIONS, OR WHEN 'STH' IS EXECUTED AS THE RESULT OF A 'PRINT	*
		5656	*	USING' CALL (SEE ERROR PROCEDURES BELOW).	*
		5657	*		*
		5658	*	*TABLES/WORK AREAS	*
		5659	*	* INTXPG (EXTERNAL IZXPAG, I\$XPAG) - 1 BYTE, FOR THE CURRENT PMC	*
		5660	*	EXECUTION PAGE NUMBER.	*
		5661	*	* INTIAR (EXTERNAL IZXIAR, I\$XIAR) - 2 BYTES, FOR THE PSEUDO	*
		5662	*	INSTRUCTION ADDRESS REGISTER.	*
		5663	*	* INTSTP (EXTERNAL IZSTAK, I\$STAK) - 2 BYTES, FOR THE RUN-TIME	*
		5664	*	STACK POINTER.	*

## S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 63
		5665	*	* INTSTI (EXTERNAL IZSTKI, ISSTKI) - 1 BYTE, FOR THE RUN-TIME	*
		5666	*	STACK POINTER INCREMENT.	*
		5667	*	* INTDT1 (EXTERNAL IZDAT1, IZDATI) - 2 BYTES, FOR THE INTERNAL	*
		5668	*	PROGRAM 'DATA' FILE 1ST ELEMENT POINTER.	*
		5669	*	* INTDAT (EXTERNAL IZDATA, I\$DATA) - 2 BYTES, FOR THE INTERNAL	*
		5670	*	PROGRAM 'DATA' FILE ELEMENT POINTER.	*
		5671	*	* INTPAR (EXTERNAL IZPARM, I\$PARM) - 2 BYTES, FOR THE INTERPRETER	*
		5672	*	COMMON PARAMETER AREA.	*
		5673	*	* INTWK1 (EXTERNAL IZWRK1, I\$WRK1) - 2 BYTES, FOR THE INTERPRETER	*
		5674	*	COMMON WORK AREA 1.	*
		5675	*	* INTWK2 (EXTERNAL IZWRK2, I\$WRK2) - 2 BYTES, FOR THE INTERPRETER	*
		5676	*	COMMON WORK AREA 2.	*
		5677	*	* INTRND (EXTERNAL IZRNSW, I\$RNSW) - 1 BYTE, FOR THE RANDOM NUMBER	*
		5678	*	INITIALIZATION SWITCH. THIS SWITCH, WHICH IS SET USING MASK	*
		5679	*	* INTRNM (EXTERNAL IZRNMK, I\$RNMK), IS USED BY THE RANDOM NUMBER	*
		5680	*	GENERATOR ('RND' FUNCTION) AS AN INDICATOR FOR 1ST FUNCTION	*
		5681	*	USAGE, AND IS INITIALIZED AT INTERPRETER ENTRY TO THE OFF STATE.	*
		5682	*	* INTERC (EXTERNAL IZCERR, I\$ERRC) - 1 BYTE, FOR THE INTERPRETER	*
		5683	*	ERROR CODE PARAMETER.	*
		5684	*	* 'PRINT USING' COMMUNICATION AREA - 12 BYTES, FOR INTER-PAGE	*
		5685	*	INFORMATION TRANSFER DURING 'PRINT USING' OPERATIONS (FZUPR).	*
		5686	*	* USER FUNCTION ACTIVITY TABLE - 22 BYTES, FOR A 'PUSH-DOWN'	*
		5687	*	STACK USED TO CONTROL RECURSIVE USER-DEFINED FUNCTION EXECUTION.	*
		5688	*	THE TABLE CONSISTS OF ELEVEN 2-BYTE ENTRY LOCATIONS, THE FIRST	*
		5689	*	OF WHICH IS SET EQUAL X'0000' TO GUARD THE BOTTOM OF THE TABLE.	*
		5690	*	EACH TABLE ENTRY LOCATION IS USED TO SAVE THE VIRTUAL ADDRESS	*
		5691	*	OF A USER FUNCTION WHICH IS IN THE PROCESS OF BEING EXECUTED	*
		5692	*	(SEE THE 'FCI' EXECUTION ROUTINE, IDIFNC).	*
		5693	*	* INTFAT (EXTERNAL IZFACT, I\$FACT) - CORE ADDRESS OF THE FIRST	*
		5694	*	BYTE IN THE USER FUNCTION ACTIVITY TABLE.	*
		5695	*	* INTFTE (EXTERNAL IZFATE, I\$FATE) - CORE ADDRESS OF THE LAST	*
		5696	*	BYTE IN THE USER FUNCTION ACTIVITY TABLE.	*
		5697	*	* INTFAP (EXTERNAL IZFATP, I\$FATP) - 2 BYTES, FOR THE FUNCTION	*
		5698	*	ACTIVITY TABLE POINTER. THIS ALWAYS REFERENCES THE LEFT BYTE	*
		5699	*	OF THE 'TOP' TABLE ENTRY.	*
		5700	*	* PMC EXECUTION BRANCH ADDRESS TABLE - 102 BYTES, FOR PMC OPCODE	*
		5701	*	TRANSLATION TO EXECUTION ROUTINE CORE ADDRESS ENTRY POINTS.	*
		5702	*	THE TABLE CONSISTS OF 51 2-BYTE EXECUTION ROUTINE ENTRY POINT	*
		5703	*	CORE ADDRESSES SEQUENCED SUCH THAT PSEUDO INSTRUCTION OPCODES	*
		5704	*	CAN BE USED TO DIRECTLY INDEX THE APPROPRIATE EXECUTION ROUTINE.	*
		5705	*	THIS TABLE, WITH FIRST BYTE REFERENCED BY LABEL INTBAT, CON-	*
		5706	*	TAINS ENTRIES FOR ALL PSEUDO INSTRUCTIONS EXCEPT THOSE WHICH	*
		5707	*	ARE DEFINED AS NON-EXECUTABLE ('DCA', 'DDL', 'DWA', 'EOF').	*
		5708	*		*
		5709	*	*ATTRIBUTES	*
		5710	*	* REUSABLE	*
		5711	*	* RELOCATABLE	*
		5712	*		*
		5713	*	*CHARACTER CODE DEPENDENCY	*
		5714	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*
		5715	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
		5716	*		*
		5717	*	*NOTES	*
		5718	*	ERROR PROCEDURES	*
		5719	*	* ERROR 1 - RUN-TIME STACK POINTER INTSTP IS INCREMENTED USING	*
		5720	*	ENTRY POINT INTADS, AND STACK CAPACITY IS EXCEEDED.	*



## S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 64
5721	*			AN ERROR CODE FOR THE MESSAGE 'EXPRESSION TOO COMPLEX TO	*
5722	*			EXECUTE' IS ESTABLISHED IN PARAMETER INTERC, AND CONTROL IS	*
5723	*			PASSED TO ERROR ROUTINE INTERR TO ABORT EXECUTION.	*
5724	*		*	ERROR 2 - AN 'STH' INSTRUCTION IS EXECUTED, AND THE 'STH'	*
5725	*			OPERAND IS IDENTICAL TO THE CURRENTLY ACTIVE LINE NUMBER IN	*
5726	*			\$INLNO. UNLESS THIS IS A VALID CONDITION, AN ERROR CODE FOR	*
5727	*			THE MESSAGE 'STATEMENT BRANCHES TO ITSELF' IS ESTABLISHED IN	*
5728	*			PARAMETER INTERC AND CONTROL IS PASSED TO ERROR ROUTINE	*
5729	*			INTERR TO ABORT EXECUTION.	*
5730	*		*	ERROR 3 - AN 'STH' INSTRUCTION IS EXECUTED, AND THE PREVI-	*
5731	*			OUSLY EXECUTED INSTRUCTION HAS A 'BNX' (SEE ICBRAN) SO THAT	*
5732	*			SWITCH INTISW IS SET ON. AN ERROR CODE FOR THE MESSAGE	*
5733	*			'NO IMAGE STATEMENT REFERENCED' IS ESTABLISHED IN PARAMETER	*
5734	*			INTERC AND CONTROL IS PASSED TO ERROR ROUTINE INTERR TO	*
5735	*			ABORT EXECUTION.	*
5736	*				*
5737	*			REGISTER USAGE	*
5738	*		*	REGISTER @BR IS NOT SAVED. IT IS USED FOR INTERP BASE	*
5739	*			ADDRESSABILITY, AND RETAINS THE INTERP BASE ADDRESS AT EXIT.	*
5740	*			SINCE INTERP CONTAINS MOST OF THE COMMONLY REFERENCED INTER-	*
5741	*			PRETER WORK AREAS AND ENTRY POINTS, CORE-RESIDENT PMC EXECU-	*
5742	*			TION ROUTINES CAN USUALLY TAKE ADVANTAGE OF THIS REGISTER	*
5743	*			CONDITION.	*
5744	*		*	REGISTER @XR IS NOT SAVED. IT IS USED AS A GENERAL PURPOSE	*
5745	*			INDEX FOR PSEUDO INSTRUCTION ACCESSING, TABLE LOOK-UP, ETC.	*
5746	*				*
5747	*			SAVED/RESTORED AREAS	*
5748	*			N/A	*
5749	*				*
5750	*			MODIFICATION CONSIDERATIONS	*
5751	*			PSEUDO INSTRUCTION EXECUTION IS BASED UPON THE SEQUENCE AND	*
5752	*			LENGTH OF THE ENTRIES IN THE EXECUTION BRANCH ADDRESS TABLE.	*
5753	*			TABLE ENTRIES ARE SELECTED BY DIRECT INDEXING USING PSEUDO	*
5754	*			INSTRUCTION OPCODES, AND THESE OPCODES ARE KEYED TO THE TABLE	*
5755	*			CONFIGURATION. ANY CHANGES TO PSEUDO INSTRUCTION OPCODES OR	*
5756	*			EXECUTION BRANCH ADDRESS TABLE ENTRY CHARACTERISTICS MUST TAKE	*
5757	*			FULL CONSIDERATION OF THIS RELATIONSHIP.	*
5758	*				*
5759	*			REQUIRED MODULES	*
5760	*		*	@SYSEQ - COMMON SYSTEM EQUATES.	*
5761	*		*	@FXDEQ - SYSTEM NUCLEUS ADDRESSES AND INDICATOR EQUATES.	*
5762	*		*	@ERMEQ - SYSTEM ERROR MESSAGE CODE EQUATES.	*
5763	*		*	\$V\$EQU - VIRTUAL MEMORY FIXED ADDRESS EQUATES.	*
5764	*		*	\$B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.	*
5765	*		*	\$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD. PREC. ONLY).	*
5766	*		*	\$I@LEQ - INTERPRETER PARAMETER EQUATES (FOR LONG PREC. ONLY).	*
5767	*		*	ICFLTA - INTERPRETER SCALAR ARITHMETIC PMC ROUTINES.	*
5768	*		*	ICMATF - INTERPRETER MATRIX FUNCTION PMC ROUTINES.	*
5769	*		*	ICELSR - INTERPRETER ELEMENT STACKING PMC ROUTINES.	*
5770	*		*	ICARST - INTERPRETER ARRAY ELEMENT STACKING PMC ROUTINES.	*
5771	*		*	ICIEST - INTERPRETER LOGICAL PMC EXECUTION ROUTINES.	*
5772	*		*	ICBRAN - INTERPRETER BRANCH PMC EXECUTION ROUTINES.	*
5773	*		*	ICLOOP - INTERPRETER 'FOR'/'NXT' PMC EXECUTION ROUTINES.	*
5774	*		*	ICVMEX - INTERPRETER INTERFACE TO V.M. RESIDENT PMC ROUTINES.	*
5775	*		*	IPGMDL - INTERPRETER PAGING CONTROL MODULE.	*
5776	*		*	IZCOMN - INTERPRETER COMMON ADDRESS REFERENCE EQUATES.	*

[illegible][illegible][illegible]

S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20 PAGE 66
				0C5C	5782	INTERP	EQU *	INTERP ENTRY POINT
				0C60	5783		USING INT010,@BR	DEFINE INTERP BASE ADDRESS
0C5C	C2	01	0C60		5784		LA INT010,@BR	LOAD INTERP BASE ADDRESS
					5785	*		
					5786	*	ESTABLISH VIRTUAL ADDRESS FOR 1ST INSTRUCTION IN NEW PMC PAGE	
					5787	*		
0C60	3C	00	1449		5788	INT010	MVI IZPGNO,*-*	SET EXECUTION PAGE NUMBER
0C61					5789		ORG INT010+@Q	INITIALIZE EXECUTION PAGE NO.
0C61	56			0C61	5790		DC AL(@VADDR-1) (@VENTA)	* (INTXPG) TO FIRST PSEUDO
0C64					5791		ORG INT010+@INST4	* CODE PAGE IN VIRTUAL MEMORY
					5792	*		
0C64	3C	00	144A		5793		MVI IZPGDS,@ZERO	SET STARTING ADDRESS OF PAGE
					5794	*		
					5795	*****		

S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 67
			5797	*****	
			5798	* ENTRY INTPAG - ESTABLISH EXECUTION CONTROL FOR TNE PSEUDO	
			5799	* INSTRUCTION REFERENCED BY PAGING ROUTINE INPUT PARAMETERS	
			5800	*****	
			5801	*	
		0C68	5802	INTPAG EQU * INTPAG ENTRY PRINT	
			5803	*	
			5804	* CORELOAD AND LOCK THE CURRENTLY REFERENCED VIRTUAL PAGE	
			5805	*	
0C68	C0 87 1354		5806	INT020 B IPGLOK LINK TO LOCK & GET INST CADDR	
			5807	*	
			5808	* RESET INSTRUCTION ADDRESS REGISTER TO RESUME EXECUTION	
			5809	*	
0C6C	4C 01 EC 144C		5810	INT030 MVC INTIAR(,@BR),IZCADR(@CADDR) SET PSEUDO IAR FOR NEXT INST	
0C71	F2 87 19		5811	J INTXEC * AND BRANCH TO EXECUTE	
			5812	*	
			5813	*****	

[illegible]

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER	15,	MOD	00	06/09/20	PAGE	68
					5815		*****							
					5816	*	ENTRY INTAD4 - INCREMENT PSEUDO IAR FOR A 4-BYTE INSTRUCTION							
					5817		*****							
					5818	*								
				0C74	5819	INTAD4 EQU	*					INTAD4 ENTRY POINT		
					5820	*								
					5821	*	INCREMENT THE INSTRUCTION ADDRESS REGISTER TO CONTINUE EXECUTION							
					5822	*								
	0C74	5E	00	EC	E1	5823	INT040 ALC	INTIAR(,@BR),INTBN4(@CADDR-1,@BR)	INCR	PAST	4-BYTE	INST		
	0C78	F2	87	12	5824		J	INTXEC	*	AND	CONTINUE	EXECUTION		
					5825	*								
					5826		*****							

S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20	PAGE 69
					5828	*****			
					5829	* ENTRY INTAD3 - INCREMENT PSEUDO IAR FOR A 3-BYTE INSTRUCTION			
					5830	*****			
					5831	*			
				0C7B	5832	INTAD3 EQU *	INTAD3 ENTRY POINT		
					5833	*			
					5834	* INCREMENT THE INSTRUCTION ADDRESS REGISTER TO CONTINUE EXECUTION			
					5835	*			
	0C7B	5E	00	EC	E0	5836	INT050 ALC INTIAR(,@BR),INTBN3(@CADDR-1,@BR) INCR PAST 3-BYTE INST		
	0C7F	F2	87	0B	5837	J INTXEC	* AND CONTINUE EXECUTION		
					5838	*			
					5839	*****			

[illegible]

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20	PAGE	70
					5841	*****				
					5842	* ENTRY INTAD2 - INCREMENT PSEUDO IAR FOR A 2-BYTE INSTRUCTION				
					5843	*****				
					5844	*				
				0C82	5845	INTAD2 EQU *			INTAD2 ENTRY POINT	
					5846	*				
					5847	* INCREMENT THE INSTRUCTION ADDRESS REGISTER TO CONTINUE EXECUTION				
					5848	*				
	0C82	5E 00 EC DF			5849	INT060 ALC INTIAR(,@BR),INTBN2(@CADDR-1,@BR) INCR PAST 2-BYTE INST				
	0C86	F2 87 04			5850	J INTXEC			* AND CONTINUE EXECUTION	
					5851	*				
					5852	*****				



S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER	15,	MOD	00	06/09/20	PAGE	71
					5854	*****								
					5855	* ENTRY	INTAD1 - INCREMENT PSEUDO IAR FOR A 1-BYTE INSTRUCTION							
					5856	*****								
					5857	*								
				0C89	5858	INTAD1 EQU	* INTAD1 ENTRY POINT							
					5859	*								
					5860	* INCREMENT	THE INSTRUCTION ADDRESS REGISTER TO CONTINUE EXECUTION							
					5861	*								
0C89	5E	00	EC	DE	5862	INT070 ALC	INTIAR(,@BR),INTBN1(@CADDR-1,@BR) INCR PAST 1-BYTE INST							
					5863	*								
					5864	*****								

S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	06/09/20	PAGE 72
					5866	*****				
					5867	* ENTRY INTXEC - ACCESS AND BRANCH TO EXECUTE CURRENT INSTRUCTION				
					5868	*****				
					5869	*				
				0C8D	5870	INTXEC EQU *	INTXEC ENTRY POINT			
					5871	*				
					5872	* ESTABLISH THE EXECUTION ADDRESS FROM INTERPRETER BRANCH TABLE				
					5873	*				
	0C8D	75	02	EC	5874	INT080 L INTIAR(, @BR), @XR	LOAD INSTRUCTION CORE ADDRESS			
	0C90	6C	00	39 00	5875	MVC INT090+@D1(, @BR), I@XOPC(B@LCOP, @XR)	SET DISP EQUAL OPCODE			
	0C94	D2	02	FC	5876	LA INTBAT-@CADDR+1(, @BR), @XR	LOAD BRANCH ADDRESS TABLE BASE			
	0C97	B5	02	00	5877	INT090 L *-*(, @XR), @XR	LOAD EXECUTION BRANCH ADDRESS			
					5878	*				
					5879	* BRANCH TO EXECUTE THE CURRENT PSEUDO INSTRUCTION				
					5880	*				
	0C9A	E0	87	00	5881	INT100 B INTXAD(, @XR)	GO, GO, EXEC CURRENT PSEUDO INST			
					5882	*				
					5883	*****				

S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 73
		5885		*****	
		5886		* ENTRY INTADS - INCREMENT THE RUN-TIME STACK POINTER	
		5887		*****	
		5888		*	
0C9D 74 08 47		0C9D 5889	INTADS EQU	* INTADS ENTRY POINT	
		5890	ST	INT120+@OP1(,@BR),@ARR SET RETURN BRANCH ADDRESS	
		5891		*	
		5892		* INCREMENT THE STACK POINTER AS SPECIFIED BY CALLING ROUTINE	
		5893		*	
0CA0 5E 00 EE EF		5894	INT110 ALC	INTSTP(,@BR),INTSTI(@CADDR-1,@BR) INCREMENT THE POINTER	
		5895		* AND RETURN TO CALLER	
0CA4 C0 82 0000		5896	INT120 BL	*-* * IF NO STACK OVERFLOW	
		5897		*	
		5898		* STACK OVERFLOW - SET 'EXPRESSION TOO COMPLEX TO EXECUTE' ERROR	
		5899		*	
0CA8 7C C7 5C		5900	INT130 MVI	INTERC(,@BR),@@E730 SET THE ERROR MESSAGE CODE	
		5901		*	
		5902		*****	

## S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 74
		5904		*****	
		5905		* ENTRY INTERR - INTERPRETER ERROR EXIT ROUTINE	
		5906		*****	
		5907		*	
		0CAB 5908	INTERR EQU *	INTERR ENTRY DONT	
		5909		*	
		5910		* CLOSE ALL ACTIVE EXTERNAL DATA FILES	
		5911		*	
0CAB C0 87 0E24		5912	INT140 B	INT700	CHECK LINE PRT CONFIGURATION
0CAF C0 87 130B		5913		B IPGCAL	LINK TO CLOSE ALL DATA FILES
0CB3 2400		0CB4 5914		DC AL(@VADDR)(V\$XKCA)	FILE CLOSING ROUTINE VADDR
		5915		*	
		5916		* PUSH ALL MODIFIED CORE PAGES TO DISK VIRTUAL MEMORY	
		5917		*	
0CB5 C0 87 130B		5918		B IPGCAL	LINK TO WRITE MODIFIED PAGES
0CB9 4C00		0CBA 5919		DC AL(@VADDR)(V\$VMPS)	VIRTUAL MEMORY PUSH RTN VADDR
		5920		*	
		5921		* ESTABLISH ERROR CODE AND EXIT TO DISPLAY THE ERROR MESSAGE	
		5922		*	
0CBB 3C 00 03CD		5923	INT150 MVI	\$CAERR,*-*	SET SYSTEM ERROR ROUTINE CODE
0CBC		5924		ORG INT150+@Q	INITIALIZE INTERPRETER ERROR
0CBC 00		0CBC 5925		DC AL(B@LCER)(I@NERR)	* CODE (INTERC) TO SPECIFY A
0CBF		5926		ORG INT150+@INST4	* NULL ERROR CONDITION
		5927		*	
0CBF 3B 01 03D1		5928		SBF \$XIND2,\$EXCMD	RESET EXECUTION MODE INDR OFF
0CC3 C0 87 0469		5929		B \$CAERK	BRANCH TO TERMINATE EXECUTION
		5930		*	
		5931		*****	

S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	06/09/20	PAGE 75
				5933		*****				
				5934	*	IMAGE STATEMENT HEADER (IMH)	PSEUDO INSTRUCTION ROUTINE			
				5935		*****				
				5936	*					
				0CC7	5937	INTIMH EQU	*			BEGIN 'IMH' EXECUTION
				5938	*					
	0CC7	7D	80 7E	5939		CLI	INT210+@Q(,@BR),@NOP			TEST ROR 'PRINT USING' CALL
	0CCA	7C	80 7E	5940		MVI	INT210+@Q(,@BR),@NOP			SET IMAGE REFERENCE SWITCH OFF
				5941	*					GO INCR PMC POINTER TO NEXT
	0CCD	D0	01 1B	5942		BNE	INTAD3(,@BR)			* INST WHEN 'PRINT USING' CALL
				5943	*					
				5944		*****				

## S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 06/09/20 PAGE 76

```

5946 *****
5947 * STATEMENT HEADER (STH) PSEUDO INSTRUCTION PROCESSING ROUTINE
5948 *****
5949 *
0CD0 5950 INTSTH EQU * BEGIN 'STH' EXECUTION
5951 *
5952 * ACCESS THE STATEMENT HEADER PSEUDO INSTRUCTION.
5953 *
0CD0 75 02 EC 5954 INT200 L INTIAR(, @BR), @XR LOAD INSTRUCTION CORE ADDRESS
0CD3 74 02 F1 5955 ST INTSHA(, @BR), @XR SAVE THE STATEMENT HEADER
0CD6 5C 00 F0 01 5956 MVC INTSHA-1(, @BR), INTXPG(1, @BR) * PMC VIRTUAL ADDRESS
5957 *
5958 * TEST FOR AN 'STH' CALL BY A 'PRINT USING' STATEMENT - EXECUTE
5959 * 'NO IMAGE STATEMENT REFERENCED' ERROR WHEN CURRENT STATEMENT HAS
5960 * BEEN REREQENCED WITH 'PRINT USING' BUT IS NOT AN IMAGE.
5961 *
0CDA 7C C2 5C 5962 MVI INTERC(, @BR), @@E725 SET IMAGE REFERENCE ERROR CODE
5963 *
0CDD D0 00 4B 5964 INT210 BC INTERR(, @BR), *- * GO TERMINATE ON ERROR WHEN
0CDE 5965 ORG INT210+@Q * IMAGE REF SWITCH IS ON (@UCB)
0CDE 80 0CDE 5966 DC AL1(@NOP) INITIALIZE IMAGE REFERENCE
0CE0 5967 ORG INT210+@INST3 * SWITCH TO OFF (@NOP) STATUS
5968 *
5969 * TEST OR A RECURSIVE TRANSFER OR CONTROL - EXECUTE 'STATEMENT
5970 * BRANCHES TO ITSELF' ERROR WHEN NEW STATEMENT NO. IS IDENTICAL TO
5971 * CURRENT STATEMENT NO. AND STATEMENT RECURSION IS NOT PERMITTED.
5972 *
0CE0 7C C6 5C 5973 MVI INTERC(, @BR), @@E729 SET STMT RECURSION ERROR CODE
0CE3 8D 01 02 03CF 5974 CLC I@XLNO(, @XR), $INLNO(B@LCLN) TEST FOR IDENTICAL STMT NOS.
5975 *
0CE8 D0 00 4B 5976 INT220 BC INTERR(, @BR), *- * GO TERMINATE ON ERR WHEN STMT
0CE9 5977 ORG INT220+@Q * NOS. IDENTICAL AND RECURSION
0CE9 81 0CE9 5978 DC AL1(@BE) * IS NOT ALLOWED (SW = @BE) -
0CEB 5979 ORG INT220+@INST3 INITLZ SW TO PROHIBIT RECURSION
5980 * (ERROR DISABLED WHEN SW = @NOP)
5981 *
0CEB 7C 81 89 5982 MVI INT220+@Q(, @BR), @BE RESET SW TO PROHIBIT RECURSION
0CEE 7C 00 5C 5983 MVI INTERC(, @BR), I@NERR RESET NULL INTERP ERROR CODE
5984 *
5985 * TEST NECESSITY TO CHECK FOR CONSOLE INTERRUPT OR STEP MODE
5986 *
0CF1 F2 00 22 5987 INT230 JC INT270, *- * BYPASS CONSOLE INTERRUPT AND
0CF2 5988 ORG INT230+@Q * STEP MODE PROCESSING WHEN
0CF2 87 0CF2 5989 DC AL1(@UCB) * STH IS 1ST PROGRAM INST OR
0CF4 5990 ORG INT230+@INST3 * FOLLOWS AN HLT INSTRUCTION
5991 *
5992 * TEST FOR AND HONOR A SUSPENDED CONSOLE INTERRUPT
5993 *
0CF4 C0 87 048D 5994 INT240 B $UNMSK LINK TO CHECK INQUIRY REQUEST
0CF8 3C 80 0476 5995 MVI $CIMSK, @NOP RESTORE CONSOLE INTERRUPT MASK
0CFC 38 02 03D1 5996 TBN $XIND2, $PAUSE TEST WHETHER INTERRUPT OCCURRED
0D00 F2 10 16 5997 JT INT280 BRANCH AFTER AN INTERRUPT
5998 *
0D03 38 02 03D0 5999 INT250 TBN $XIND1, $STEPT TEST FOR STEP MODE EXECUTION
0D07 F2 90 0F 6000 JF INT280 BRANCH IF NOT IN STEP MODE
6001 *

```

## S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	06/09/20	PAGE 77
					6002	*	STEP MODE - RETURN TEMPORARILY TO SYSTEM AFTER STATEMENT EXECUTION			
					6003	*				
	0D0A	3A	06	03D1	6004	INT260	SBN \$XIND2,\$PAUSE+\$PSTEP			SET STEP MODE INDICATOR
	0D0E	C0	87	0E24	6005		B INT700			CHECK LINE PRT CONFIGURATION
	0D12	C0	87	04BA	6006		B \$PAUSD			LINK TO PAUSE IN STEP MODE
					6007	*				
					6008	*	ENABLE CONSOLE INTERRUPT AND POSSIBLE STEP MODE PROCESSING			
					6009	*				
	0D16	7C	80	92	6010	INT270	MVI INT230+@Q(,@BR),@NOP			SET THE CI/STEP DISABLER OFF
					6011	*				
					6012	*	TEST FOR A USER-REQUESTED PROGRAM EXECUTION TERMINATION			
					6013	*				
	0D19	38	10	03D1	6014	INT280	TBN \$XIND2,\$ABORT			IC EXECUTION HAS BEEN ABORTED
	0D1D	C0	10	0E0C	6015		BT INTSVC			T GO EXECUTE SUPERVISOR CALL
					6016	*				
					6017	*	RESET ALL PAUSE CONDITION SYSTEM INDICATORS			
					6018	*				
	0D21	3B	0E	03D1	6019	INT285	SBF \$XIND2,\$PAUSE+\$PSTEP+\$PSTMT			SET ALL PAUSE INDICATORS OFF
					6020	*				
					6021	*	STORE THE STH INSTRUCTION OPERAND AS THE NEW STATEMENT NUMBER			
					6022	*				
	0D25	2C	01	03CF 02	6023	INT290	MVC \$INLNO,I@XLNO(B@LCLN,@XR)			SAVE THE NEW STATEMENT NUMBER
					6024	*				
					6025	*	TEST FOR FLOW TRACE EXECUTION MODE			
					6026	*				
	0D2A	D0	00	1B	6027	INT300	BC INTAD3(,@BR),*-*			IF NOT IN FLOW TRACE MODE
	0D2B				6028		ORG INT300+@Q			* GO EXECUTE NEXT PSEUDO INST -
	0D2B	87			6029		DC AL1(@UCB)			* INITIALIZE BRANCH CONDITION
	0D2D				6030		ORG INT300+@INST3			* TO SUPPRESS FLOW TRACE
					6031	*				
					6032	*	FLOW TRACE MODE - TEST FOR ENABLED TRACE CONDITION			
					6033	*				
	0D2D	38	04	03D0	6034	INT310	TBN \$XIND1,\$TRACE			IF EXECUTION TRACE NOT ENABLED
	0D31	D0	90	1B	6035		BF INTAD3(,@BR)			* GO EXECUTE NEXT PSEUDO INST
					6036	*				
					6037	*	TRACE ENABLED - DISPLAY THE NEW STATEMENT NUMBER			
					6038	*				
	0D34	C0	87	130B	6039	INT320	B IPGCAL			LINK TO DISPLAY STMNT NUMBER
	0D38	4600			6040		DC AL(@VADDR)(V\$DTLN)			FLOW TRACE ROUTINE VADDR
					6041	*				
	0D3A	D0	87	1B	6042		B INTAD3(,@BR)			GO EXECUTE NEXT PSEUDO INST
					6043	*				
					6044	*	*****			



## S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 06/09/20 PAGE 78

```

6046 *****
6047 * INTERPRETER EXECUTIVE ROUTINE CONSTANTS
6048 *****
6049 *
0D3D 0001      0D3E 6050 INTBN1 DC      IL2'1'          BINARY INTEGER +1
0D3F 02        0D3F 6051 INTBN2 DC      IL1'2'          BINARY INTEGER +2
0D40 03        0D40 6052 INTBN3 DC      IL1'3'          BINARY INTEGER +3
0D41 04        0D41 6053 INTBN4 DC      IL1'4'          BINARY INTEGER +4
0D42 FFFF      0D43 6054 INTBM1 DC      IL2'-1'         BINARY INTEGER -1
6055 *
0D44 08        0D44 6056 INTL1F DC      AL1(I@LUFV)      LENGTH OF 1 FLOATING PT VALUE
0D45 10        0D45 6057 INTL2F DC      AL1(2*I@LUFV)    LENGTH OF 2 FLOATING PT VALUES
0D46 13        0D46 6058 INTL1C DC      AL1(I@LCRV)      LENGTH OF 1 CHARACTER VALUE
0D47 26        0D47 6059 INTL2C DC      AL1(2*I@LCRV)    LENGTH OF 2 CHARACTER VALUES
0D48 0002      0D49 6060 INTLVA DC      AL2(@VADDR)      LENGTH OF A VIRTUAL ADDRESS
0D4A 0A        0D4A 6061 INTLFA DC      AL1(I@LUFV+@VADDR) LENGTH OF FLT PT VALUE & VADDR
6062 *
6063 *****
6064 * INTERPRETER EXECUTIVE ROUTINE WORK AREAS
6065 *****
6066 *
0D4B          0D4C 6067 INTIAR DS      CL(@CADDR)        PSEUDO INSTRUCTION CORE ADDR
6068 *
0D4D          0D4E 6069 INTSTP DS      CL(@CADDR)        RUN-TIME STACK POINTER -
0D4D          6070          ORG      *-@CADDR            * INITIALIZE STACK POINTER
0D4D 0639      0D4E 6071          DC      AL(@CADDR)(IZSTKB) * TO REFERENCE BOTTOM OF STACK
6072 *
0D4F          0D4F 6073 INTSTI DS      CL(@CADDR-1)      RUN-TIME STACK POINTER INCR
6074 *
0D50          0D51 6075 INTSHA DS      CL(@VADDR)        STATEMENT HEADER VIRTUAL ADDR
6076 *
0D52          0D53 6077 INTDAT DS      CL(@VADDR)        INTERNAL DATA FILE POINTER
0D54          0D55 6078 INTDT1 DS      CL(@VADDR)        DATA FILE 1ST ELEMENT VADDR
6079 *
0D56          0D57 6080 INTPAR DS      CL2                INTERPRETER COMMON PARAMETER
6081 *
0D58          0D59 6082 INTWK1 DS      CL2                GENERAL PURPOSE WORK AREA 1
0D5A          0D5B 6083 INTWK2 DS      CL2                GENERAL PURPOSE WORK AREA 2
6084 *
0D5C          0D5C 6085 INTRND DS      CL1                RANDOM NUMBER INITLZN SWITCH
0D5C          6086          ORG      INTRND              INITIALIZE RANDOM NUMBER SWITCH
0D5C 00        0D5C 6087          DC      XL1'00'         TO OFF STATUS AT EXEC START
0001 6088 INTRNM EQU      X'01'          RANDOM NUMBER INITZN SW MASK
6089 *          * SW ON = PRIOR RND FUNC USAGE
6090 *
6091 *****

```

## S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 06/09/20 PAGE 79

```

6093 *****
6094 * INTERPRETER OPCODE EXECUTION BRANCH ADDRESS TABLE
6095 *****
6096 *
0D5D 6097 INTBAT EQU * BRANCH TABLE CORE ADDRESS
6098 *
0D5D 0E0C 0D5E 6099 DC AL(@CADDR)(INTSVC) SVC(X'02') SUPERVISOR CALL
0D5F 0DE9 0D60 6100 DC AL(@CADDR)(INTHLT) SLT(X'04') HALT EXECUTION
0D61 0E66 0D62 6101 DC AL(@CADDR)(ICFADD) ADD(X'06') ADD
0D63 0E5F 0D64 6102 DC AL(@CADDR)(ICFSUB) SUB(X'08') SUBTRACT
0D65 0E58 0D66 6103 DC AL(@CADDR)(ICFMPY) MPY(X'0A') MULTIPLY
0D67 0E51 0D68 6104 DC AL(@CADDR)(ICFDIV) DIV(X'0C') DIVIDE
0D69 0E32 0D6A 6105 DC AL(@CADDR)(ICFPWR) PWR(X'0E') EXPONENTIATE
0D6B 0E3F 0D6C 6106 DC AL(@CADDR)(ICFNEG) NEG(X'10') NEGATE
0D6D 0E7E 0D6E 6107 DC AL(@CADDR)(ICFFN0) FN0(X'12') FUNC CALL, 0 ARG
0D6F 0E88 0D70 6108 DC AL(@CADDR)(ICFFN1) FN1(X'14') FUNC CALL, 1 ARG
0D71 127B 0D72 6109 DC AL(@CADDR)(ICVFCI) FCI(X'16') FUNC CALL, INDIR
0D73 0EBE 0D74 6110 DC AL(@CADDR)(ICMMF1) MF1(X'18') FUNC CALL, 1 MATR
0D75 0EB9 0D76 6111 DC AL(@CADDR)(ICMMF2) MF2(X'1A') FUNC CALL, 2 MATR
0D77 0EB4 0D78 6112 DC AL(@CADDR)(ICMMF3) MF3(X'1C') FUNC CALL, 3 MATR
0D79 0EAD 0D7A 6113 DC AL(@CADDR)(ICMSM) MSM(X'1E') FUNC CALL, M/S MPY
0D7B 0F17 0D7C 6114 DC AL(@CADDR)(ICESTF) STF(X'20') STACK FLOATING VAL
0D7D 0F9D 0D7E 6115 DC AL(@CADDR)(ICASF1) SF1(X'22') STACK VECTOR VALUE
0D7F 0FA4 0D80 6116 DC AL(@CADDR)(ICASF2) SF2(X'24') STACK MATRIX VALUE
0D81 0F6A 0D82 6117 DC AL(@CADDR)(ICEUSF) USF(X'26') UNSTACK FLOATING
0D83 0F09 0D84 6118 DC AL(@CADDR)(ICESTC) STC(X'28') STACK CHAR VALUE
0D85 0FBC 0D86 6119 DC AL(@CADDR)(ICASC1) SC1(X'2A') STACK CHAR ARRY VAL
0D87 0F37 0D88 6120 DC AL(@CADDR)(ICEUSC) USC(X'2C') UNSTACK CHAR VALUE
0D89 128B 0D8A 6121 DC AL(@CADDR)(ICVSDN) SD0(X'2E') STACK DOPE VECTOR
0D8B 128B 0D8C 6122 DC AL(@CADDR)(ICVSDN) SDI(X'30') STACK D/V, REDIM 1
0D8D 128B 0D8E 6123 DC AL(@CADDR)(ICVSDN) SD2(X'32') STACK D/V, REDIM 2
0D8F 0EDB 0D90 6124 DC AL(@CADDR)(ICESTA) STA(X'34') STACK VIRTUAL ADDR
0D91 0F84 0D92 6125 DC AL(@CADDR)(ICASA1) SA1(X'36') STACK VECTOR VADDR
0D93 0F8B 0D94 6126 DC AL(@CADDR)(ICASA2) SA2(X'38') STACK MATRIX VADDR
0D95 0F92 0D96 6127 DC AL(@CADDR)(ICASB1) SBI(X'3A') STACK CHAR ARR VADR
0D97 0EF2 0D98 6128 DC AL(@CADDR)(ICESTX) STX(X'3C') STACK EXEC CRTL CODE
0D99 1102 0D9A 6129 DC AL(@CADDR)(ICTCSA) CSA(X'3E') COMPUTE STKD VADDR
0D9B 10BF 0D9C 6130 DC AL(@CADDR)(ICTCMF) CMF(X'40') COMPARE FLOATING
0D9D 10AB 0D9E 6131 DC AL(@CADDR)(ICTCMC) CMC(X'42') COMPARE CHARACTER
0D9F 1172 0DA0 6132 DC AL(@CADDR)(ICBBRC) BRC(X'44') BRANCH ON COND
0DA1 1180 0DA2 6133 DC AL(@CADDR)(ICBBRA) BRA(X'46') BRANCH UNCOND
0DA3 1153 0DA4 6134 DC AL(@CADDR)(ICBBRD) BRD(X'48') BRANCH & DELETE
0DA5 1149 0DA6 6135 DC AL(@CADDR)(ICBBNX) BNX(X'4A') BRANCH & SKIP EXEC
0DA7 115B 0DA8 6136 DC AL(@CADDR)(ICBBRS) BRS(X'4C') BRANCH STKD VADDR
0DA9 11C0 0DAA 6137 DC AL(@CADDR)(ICLFOR) FOR(X'4E') BEGIN 'FOR' LOOP
0DAB 11DC 0DAC 6138 DC AL(@CADDR)(ICLNXT) NXT(X'50') CONTINUE 'FOR' LOOP
0DAD 129A 0DAE 6139 DC AL(@CADDR)(ICVFIO) GET(X'52') TOLTT DATA ITEM
0DAF 129A 0DB0 6140 DC AL(@CADDR)(ICVFIO) PUT(X'54') OUTPUT DATA ITEM
0DB1 129A 0DB2 6141 DC AL(@CADDR)(ICVFIO) INI(X'56') INITIATE 'INPUT'
0DB3 129A 0DB4 6142 DC AL(@CADDR)(ICVFIO) ADF(X'58') ACTIVATE FILE
0DB5 129A 0DB6 6143 DC AL(@CADDR)(ICVFIO) RSR(X'5A') RESTORE DATA PT
0DB7 129A 0DB8 6144 DC AL(@CADDR)(ICVFIO) RST(X'5C') RESET FILE PT
0DB9 129A 0DBA 6145 DC AL(@CADDR)(ICVFIO) CLS(X'5E') CLOSE FILE
0DBB 129A 0DBC 6146 DC AL(@CADDR)(ICVFIO) PRS(X'60') PRINT & SPACE
0DBD 129A 0DBE 6147 DC AL(@CADDR)(ICVFIO) PRU(X'62') PRINT USING
0DBF 0CD0 0DC0 6148 DC AL(@CADDR)(INTSTH) STH(X'64') STATEMENT HEADER

```

[illegible][illegible]

## S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20 PAGE 81
		6154			*****	
		6155			* SPECIAL WORK AREAS FOR 'PRINT USING' OPERATION	
		6156			*****	
		6157			*	
0DC5		0DC5	6158	INTPIN DS	CL1 PRINT USING INDICATOR BYTE	
0DC6		0DC6	6159	INTPIL DS	CL1 IMAGE ASSEMBLE BITE LENGTH	
0DC7		0DC8	6160	INTPB1 DS	CL (@CADDR) IMAGE BUFFER 1 CORE ADDRESS	
0DC9		0DCA	6161	INTPB2 DS	CL (@CADDR) IMAGE BUFFER 2 CORE ADDRESS	
0DCB		0DCC	6162	INTPIP DS	CL (@CADDR) IMAGE SCAN POINTER	
0DCD		0DCE	6163	INTPC1 DS	CL (@CADDR) IMAGE CONV SPEC 1ST CHAR PT	
0DCF		0DD0	6164	INTPDP DS	CL (@CADDR) IMAGE CONV SPEC DECIMAL POINT PT	
		6165		*		
		0D5A	6166	INTPCC EQU	INTWK2-1 IMAGE CONV SPEC COUNTERS	
		0D58	6167	INTPSC EQU	INTPCC-2 IMAGE CONV SPEC CHAR COUNT	
		0D59	6168	INTPDC EQU	INTPCC-1 IMAGE CONV SPEC DIGIT COUNT	
		0D5A	6169	INTPFC EQU	INTPCC-0 IMAGE CONV SPEC FRACTION COUNT	
		0D5B	6170	INTPIC EQU	INTPCC+1 IMAGE CONV SPEC INTEGER COUNT	
		6171		*		
		0D56	6172	INTPJX EQU	INTPAR-1 ADJUSTED EXPONENT FOR E-FORMAT	
		6173		*		
		6174			*****	

S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20 PAGE 82
		6176			*****	
		6177			* USER FUNCTION EXECUTION ACTIVITY TABLE	
		6178			*****	
		6179			*	
		000A	6180	INTNFA EQU 10	MAXIMUM NUMBER OF ACTIVE FUNCS	
		6181			*	
		0DD1	6182	INTFAT EQU *	FUNCTION ACTIVITY TBL BASE ADDR	
0DD1 0000		0DD2	6183	DC XL(@VADDR)'00'	FUNC ACTIVITY TBL DUMMY ENTRY	
0DD3		0DE6	6184	DS CL(INTNFA*@VADDR)	FUNC ACTIVITY TABLE AREA	
		0DE6	6185	INTFTE EQU *-1	FUNC ACTIVITY TBL ENDING ADDR	
		6186			*	
		0DE7	6187	INTFAP DS CL(@CADDR)	FUNCTION ACTIVITY TABLE POINTER	
		0DE7	6188	ORG *-@CADDR	INITIALIZE THE POINTER TO	
		0DE7 0DD1	6189	DC AL(@CADDR)(INTFAT)	* REFERENCE O'TH TABLE ENTRY	
		6190			*	
		6191			*****	

S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20 PAGE 83
			6193		*****	
			6194		* INTERPRETER EXECUTIVE ROUTINE MISCELLANEOUS EQUATES	
			6195		*****	
			6196		*	
		0000	6197	INTXAD EQU 0	DISP FOR EXECTION BRANCH ADDR	
			6198		*	
		0C61	6199	INTXPG EQU INT010+@Q	CURRENT EXECUTION PAGE NUMBER	
		0CBC	6200	INTERC EQU INT150+@Q	EXECUTION ERROR MESSAGE CODE	
			6201		*	
		0CDE	6202	INTISW EQU INT210+@Q	IMAGE REFERENCE SWITCH	
			6203		* @UCB = 'PRINT USING' CALL	
			6204		* @NOP = NO 'PRINT USING' CALL	
		0CE9	6205	INTRSW EQU INT220+@Q	STATEMENT RECURSION ERR SWITCH	
			6206		* @BE = RECURSION NOT ALLOWED	
			6207		* @NOP = RECURSION PERMITTED	
		0D2B	6208	INTTSW EQU INT300+@Q	INTERPRETER TRACE FLOW SWITCH	
			6209		* @UCB = TRACE FLOW DISABLED	
			6210		* @NOP = TRACE FLOW ENABLED	
			6211		*	
			6212		*****	

S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 84
			6214		*****	
			6215		* HALT (HLT) PSEUDO INSTRUCTION PROCESSING ROUTINE	
			6216		*****	
			6217		*	
0DE9	C0 87 0E24		0DE9	6218	INTHLT EQU * BEGIN 'HLT' EXECUTION	
				6219	B INT700 CHECK LINE PRT CONFIGURATION	
			6220		*	
			6221		* RETURN TEMPORARILY TO SYSTEM FOR A PROGRAMMED PAUSE	
			6222		*	
0DED	3A 0A 03D1			6223	INT400 SBN \$XIND2,\$PAUSE+\$PSTMT SET PAUSE STATEMENT INDICATOR	
0DF1	C0 87 04BA			6224	B \$PAUSD LINK TO PAUSE IN PAUSE MODE	
			6225		*	
			6226		* DISABLE CONSOLE INTERRUPT AND POSSIBLE STEP MODE PROCESSING	
			6227		*	
0DF5	7C 87 92			6228	INT410 MVI INT230+@Q(,@BR),@UCB SET THE CI/STEP DISABLER ON	
			6229		*	
0DF8	D0 87 29			6230	B INTAD1(,@BR) GO EXECUTE NEXT PSEUDO INST	
			6231		*	
			6232		*****	



S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	06/09/20	PAGE 85
					6234	*****				
					6235	* END-OF-PAGE (EOP) PSEUDO INSTRUCTION PROCESSING ROUTINE				
					6236	*****				
					6237	*				
				0DFB	6238	INTEOP EQU *	BEGIN 'EOP' EXECUTION			
					6239	*				
					6240	* UNLOCK THE CURRENT PSEUDO INSTRUCTION PAGE FROM CORE V.M.				
					6241	*				
		0DFB	1C 00 1449 01		6242	INT500 MVC IZPGNO,INTXPG(@VADDR-1,@BR)	RESTORE CURRENT PMC PAGE NO.			
		0E00	C0 87 1350		6243	B IPGULK	LINK TO RELEASE CURRENT PAGE			
					6244	*				
					6245	* INCREMENT THE EXECUTION PAGE NUMBER AND CONTINUE PMC PROCESSING				
					6246	*				
		0E04	5E 00 01 DE		6247	INT510 ALC INTXPG(,@BR),INTBN1(@VADDR-1,@BR)	INCREMENT PMC PAGE NO.			
		0E08	C0 87 0C60		6248	B INT010	* AND GO GET NEW PAGE			
					6249	*				
					6250	*****				

## S/3 BASIC INTERPRETER EXECUTIVE ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	06/09/20	PAGE 86
					6252	*****				
					6253	* SUPERVISOR CALL (SVC) PSEUDO INSTRUCTION PROCESSING ROUTINE				
					6254	*****				
					6255	*				
0E0C	C0	87	0E24	0E0C	6256	INTSVC EQU *	BEGIN 'SVC' EXECUTION			
					6257	B INT700	CHECK LINE PRINTER CONFIG.		1-3	
					6258	*				
					6259	* CLOSE ALL ACTIVE EXTERNAL FILES				
					6260	*				
0E10	C0	87	130B		6261	INT600 B IPGCAL	LINK TO CLOSE ALL DATA FILES			
0E14	2400			0E15	6262	DC AL(@VADDR)(V\$XKCA)	FILE CLOSING ROUTINE VADDR			
					6263	*				
					6264	* PUSH ALL MODIFIED CORE PAGES TO DISK VIRTUAL MEMORY				
					6265	*				
0E16	C0	87	130B		6266	B IPGCAL	LINK TO WRITE MODIFIED PAGES			
0E1A	4C00			0E1B	6267	DC AL(@VADDR)(V\$VMPS)	VIRTUAL MEMORY PUSH RTN VADDR			
					6268	*				
					6269	* TERMINATE EXECUTION AND RETURN CONTROL TO THE SYSTEM				
					6270	*				
0E1C	3B	01	03D1		6271	INT610 SBF \$XIND2,\$EXCMD	RESET EXECUTION MODE INDR OFF			
0E20	C0	87	04A1		6272	B \$CARPL	EXYIT THE INTERPRETER			
					6274	*****				
					6275	* CHECK IF LINE PRINTER BUFFER EMPTY, PRINT IT IF NOT				
					6276	*****				
					6277	*				
0E24	34	08	0E31		6278	INT700 ST INT710+@OP1,@ARR	SAVE RETURN ADDR			
0E28	C0	87	12B1		6279	B I\$CALL	BRANCH TO CALL ROUTINE		1-4	
0E2C	24AD			0E2D	6280	DC AL(@VADDR)(V\$XKLP)	LINE PRINTER CLOSE OUT RTN.		1-4	
0E2E	C0	87	0000		6281	INT710 B *-*	RETURN TO CALLER			
					6282	*				
					6283	* END OF BASIC INTERPRETER EXECUTIVE ROUTINE CODING				
					6284	*				

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 87
		6286		*****	
		6287	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		6288	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
		6289	*		*
		6290		*****	
		6291	*	*STATUS	*
		6292	*	VERSION 1 MODIFICATION 0	*
		6293	*		*
		6294	*	*FUNCTION	*
		6295	*	* ICFLTA CONTAINS THE RUN-TIME ROUTINES WHICH INTERPRET AND CAUSE	*
		6296	*	ENECUTION OF THE FOLLOWING PSEUDO MACHINE INSTRUCTIONS -	*
		6297	*	* 'ADD' - ADD	*
		6298	*	* 'SUB' - SUBTRACT	*
		6299	*	* 'MPY' - MULTIPLY	*
		6300	*	* 'DIV' - DIVIDE	*
		6301	*	* 'PWR' - EXPONENTIATE	*
		6302	*	* 'NEG' - NEGATE	*
		6303	*	* 'FN0' - FUNCTION CALL, NO ARGUMENT	*
		6304	*	* 'FN1' - FUNCTION CALL, ONE ARGUMENT	*
		6305	*	* THE FOLLOWING DESCRIPTIONS GIVE FUNCTIONAL SPECIFICATIONS FOR	*
		6306	*	THE ROUTINES WHICH ARE USED TO EXECUTE THE PSEUDO INSTRUCTIONS	*
		6307	*	LISTED ABOVE. THESE INSTRUCTIONS INVOLVE ARITHMETIC OPERATIONS	*
		6308	*	IN THE RUN-TIME STACK. AND ALL REFERENCES TO ARITHMETIC VALUES	*
		6309	*	IMPLY UNPACKED FLOATING POINT DECIMAL ELEMENTS.	*
		6310	*	* 'ADD' - ADD (FORMAT - OP)	*
		6311	*	THE FLOATING POINT VALUE AT THE TOP OF THE STACK IS ADDED	*
		6312	*	TO THE SECOND VALUE IN THE STACK. BOTH VALUES ARE DELETED	*
		6313	*	FROM THE STACK AND THE SUM IS PLACED AT THE TOP OF THE	*
		6314	*	STACK.	*
		6315	*	* 'SUB' - SUBTRACT (FORMAT - OP)	*
		6316	*	THE FLOATING POINT VALUE AT THE TOP OF THE STACK IS SUB-	*
		6317	*	TRACTED FROM THE SECOND VALUE IN THE STACK. BOTH VALUES	*
		6318	*	ARE DELETED FROM THE STACK AND THE DIFFERENCE IS PLACED AT	*
		6319	*	THE TOP OF THE STACK.	*
		6320	*	* 'MPY' - MULTIPLY (FORMAT - OP)	*
		6321	*	THE FLOATING POINT VALUE SECOND IN THE STACK IS MULTIPLIED	*
		6322	*	BY THE VALUE AT THE TOP OF THE STACK. BOTH VALUES ARE	*
		6323	*	DELETED FROM THE STACK AND THE PRODUCT IS PLACED AT THE	*
		6324	*	TOP OF THE STACK.	*
		6325	*	* 'DIV' - DIVIDE (FORMAT - OP)	*
		6326	*	THE FLOATING POINT VALUE SECOND IN THE STACK IS DIVIDED BY	*
		6327	*	THE VALUE AT THE TOP OF THE STACK. BOTH VALUES ARE DE-	*
		6328	*	LETED FROM THE STACK AND THE QUOTIENT IS PLACED AT THE TOP	*
		6329	*	OF THE STACK.	*
		6330	*	* 'PWR' - EXPONENTIATE (FORMAT - OP)	*
		6331	*	THE FLOATING POINT VALUE SECOND IN THE STACK IS RAISED TO	*
		6332	*	THE POWER SPECIFIED BY THE VALUE AT THE TOP OF THE STACK.	*
		6333	*	BOTH VALUES ARE DELETED FROM THE STACK AND THE RESULT IS	*
		6334	*	PLACED AT THE TOP OF THE STACK.	*
		6335	*	* 'NEG' - NEGATE (FORMAT - OP)	*
		6336	*	THE FLOATING POINT VALUE AT THE TOP OF THE STACK IS	*
		6337	*	NEGATED. THE VALUE AT THE TOP OF THE STACK IS DELETED AND	*
		6338	*	THE NEGATED VALUE IS PLACED AT THE TOP OF THE STACK.	*
		6339	*	* 'FN0' - FUNCTION CALL, NO ARGUMENT (FORMAT - OP VADR)	*
		6340	*	NO ARGUMENT IS REQUIRED FOR THE FUNCTION ROUTINE WHOSE	*
		6341	*	ENTRY ADDRESS IS VADR. THE FUNCTION RESULT IS PLACED AT	*

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 06/09/20 PAGE 88

```

6342 *          THE TOP OF THE STACK.
6343 *          * 'FN1' - FUNCTION CALL, ONE ARGUMENT (FORMAT - OP VADR)
6344 *          THE FLOATING POINT VALUE AT THE TOP OF THE STACK IS USED
6345 *          AS THE ARGUMENT FOR THE FUNCTION WHOSE ENTRY ADDRESS IS
6346 *          VADR. THE VALUE AT THE TOP OF THE STACK IS DELETED AND
6347 *          THE FUNCTION RESULT IS PLACED AT THE TOP OF THE STACK.
6348 *
6349 *ENTRY POINTS
6350 *          * ENTRY ICFADD - FOR EXECUTION OF THE 'ADD' INSTRUCTION.
6351 *          CALLING SEQUENCE IS
6352 *          B          ICFADD
6353 *          * ENTRY ICFSUB - FOR EXECUTION OF THE 'SUB' INSTRUCTION.
6354 *          CALLING SEQUENCE IS
6355 *          B          ICFSUB
6356 *          * ENTRY ICEMPY - FOR EXECUTION OF THE 'MPY' INSTRUCTION.
6357 *          CALLING SEQUENCE IS
6358 *          B          ICFMPY
6359 *          * ENTRY ICFDIV - FOR EXECUTION OF THE 'DIV' INSTRUCTION.
6360 *          CALLING SEQUENCE IS
6361 *          B          ICFDIV
6362 *          * ENTRY ICFPWR - FOR EXECUTION OF THE 'PWR' INSTRUCTION.
6363 *          CALLING SEQUENCE IS
6364 *          B          ICFPWR
6365 *          * ENTRY ICENEG - FOR EXECUTION OF THE 'NEG' INSTRUCTION.
6366 *          CALLING SEQUENCE IS
6367 *          B          ICFNEG
6368 *          * ENTRY ICFFN0 - FOR EXECUTION OF THE 'FN0' INSTRUCTION.
6369 *          CALLING SEQUENCE IS
6370 *          B          ICFFN0
6371 *          * ENTRY ICFFN1 - FOR EXECUTION OF THE 'FN1' INSTRUCTION.
6372 *          CALLING SEQUENCE IS
6373 *          B          ICFFN1
6374 *          * EACH OF THE ABOVE ENTRY POINTS IS ACCESSED THROUGH THE PMC
6375 *          EXECUTION BRANCH ADDRESS TABLE (INTBAT) IN EXECUTIVE ROUTINE
6376 *          INTERP, AND IS SUBJECT TO INPUT CONDITIONS DESCRIBED BELOW.
6377 *
6378 *INPUT
6379 *          * IZSTAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER. THIS IS TO
6380 *          CONTAIN THE CORE ADDRESS OF THE STACK LOCATION IMMEDIATELY
6381 *          FOLLOWING THE LAST STACKED VALUE.
6382 *          * IZXIAR - 2 BYTES, FOR THE PMC ADDRESS REGISTER. THIS IS TO
6383 *          CONTAIN THE CORE ADDRESS OF THE OPCODE FIELD IN THE PSEUDO
6384 *          INSTRUCTION BEING EXECUTED.
6385 *          * RUN-TIME STACK (FOR ENTRY POINTS ICFADD, ICFSUB, ICFMPY, ICEDIV,
6386 *          ICFPWR) - THIS CONTAINS TWO FLOATING POINT VALUES - ONE AT THE
6387 *          TOP AND ONE AT THE SECOND STACK POSITIONS.
6388 *          * RUN-TIME STACK (FOR ENTRY POINTS ICFNEG, ICFFN1) - THIS CONTAINS
6389 *          A FLOATING POINT VALUE AT THE TOP STACK POSITION.
6390 *          * RUN-TIME STACK (FOR ENTRY POINT ICFFN0) - THIS CONTAINS NO
6391 *          INPUT ARGUMENT.
6392 *
6393 *OUTPUT
6394 *          * IZSTAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER. THIS CON-
6395 *          TAINS THE CORE ADDRESS OF THE STACK LOCATION IMMEDIATELY
6396 *          FOLLOWING THE STACKED RESULTING VALUE.
6397 *          * IZERRC - 1 BYTE FOR THE ERROR CONDITION CODE, THIS CONTAINS A

```

S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 89
		6398	*	NULL CODE (I@NERR) WHEN NO ERROR CONDITION EXISTS, OR AN ERROR	*
		6399	*	CODE SPECIFYING THE PARTICULAR ERROR CONDITION DISCOVERED.	*
		6400	*	* IZSLLC (AFTER ENTRY POINT ICEFN0) - 1 BYTE, FOR THE STACKED	*
		6401	*	VALUE LENGTH CODE. THIS IS SET TO CONTAIN A VALUE WHICH IS ONE	*
		6402	*	LESS THAN THE PACKED FLOATING POINT VALUE LENGTH, SIMULATING	*
		6403	*	STACKING ROUTINE ISTACK'S OPERATION (ISTACK IS NOT USED BY THE	*
		6404	*	FUNCTION WHICH STACKS THE VALUE).	*
		6405	*	* RUN-TIME STACK - THIS CONTAINS THE RESULTING FLOATING POINT	*
		6406	*	VALUE AT THE TOP STACK POSITION.	*
		6407	*		*
		6408	*	*EXTERNAL REFERENCES	*
		6409	*	* FDIADD - ENTRY POINT FOR FLOATING POINT ADD ROUTINE.	*
		6410	*	* FOISUB - ENTRY POINT FOR FLOATING POINT SUBTRACT ROUTINE.	*
		6411	*	* FZIMPY - ENTRY POINT FOR FLOATING POINT MULTIPLY ROUTINE.	*
		6412	*	* FFIDVD - ENTRY POINT FOR FLOATING POINT DIVIDE ROUTINE.	*
		6413	*	* V\$APWR - VIRTUAL ENTRY ADDRESS FOR FNBPR, FLT, PT. POWER RTN.	*
		6414	*	* IDGCAL - ENTRY POINT FOR PAGING MODULE V.M. PROGRAM CALL RTN.	*
		6415	*	* INTADS - ENTRY POINT FOR INTERPRETER STACK POINTER INCREMENTER.	*
		6416	*	* INTAD1 - ENTRY POINT FOR INTERPRETER 1-BYTE PMC INCREMENT RTN.	*
		6417	*	* INTAD3 - ENTRY POINT FOR INTERPRETER 3-BYTE PMC INCREMENT RTN.	*
		6418	*	* INTERR - ENTRY POINT FOR INTERPRETER EXECUTION ERROR ROUTINE.	*
		6419	*	* IZBASE - CORE ADDRESS FOR INTERP BASE ADDRESSABILITY.	*
		6420	*	* IZSTAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER.	*
		6421	*	* IZYIAR - 2 BYTES, FOR THE PSEUDO INSTRUCTION ADDRESS REGISTER.	*
		6422	*	* IZPARM - 2 BYTES, FOR THE INTERPRETER COMMUNICATION PARAMETER.	*
		6423	*	THIS IS USED IN ICFLTA TO DEFINE WHETHER AN 'FN0' OR 'FM1' CALL	*
		6424	*	IS BEING MADE TO A VIRTUAL MEMORY EXECUTION SUBROUTINE.	*
		6425	*	* IZSTKI - 1 BYTE, FOR THE STACK INCREMENT PARAMETER TO INTADS.	*
		6426	*	* IZERRC - 1 BYTE, FOR THE INTERPRETER EXECUTION ERROR CODE.	*
		6427	*	* IZCL1F - 1 BYTE, FOR LENGTH OF AN UNPACKED FLOATING POINT VALUE.	*
		6428	*	* IICL2F - 1 BYTE. FOR LENGTH OF 2 UNPACKED FLOATING POINT VALUES.	*
		6429	*	* IZSLLC - 1 BYTE, FOR THE STACKED VALUE LENGTH CODE (SEE ISTACK).	*
		6430	*		*
		6431	*	*EXITS, NORMAL	*
		6432	*	* ENTRY POINTS ICFFN0,ICFFN1 - CONTROL IS NORMALLY PASSED TO THE	*
		6433	*	INTERPRETER EXECUTIVE AT ENTRY POINT INTAD3 FOR NEXT PSEUDO	*
		6434	*	INSTRUCTION EXECUTION.	*
		6435	*	* ALL OTHER ENTRY POINTS - CONTROL IS NORMALLY PASSED TO THE	*
		6436	*	INTERPRETER EXECUTIVE AT ENTRY POINT INTADI FOR NUT PSEUDO	*
		6437	*	INSTRUCTION EXECUTION.	*
		6438	*		*
		6439	*	*EXITS, ERROR	*
		6440	*	ALL ENTRY POINTS - CONTROL IS PASSED TO THE INT:RPRETEQ EXECUTIVE	*
		6441	*	AT ENTRY POINT INTERR WITH PARAMETER IZERRC CONTAINING THE APPRO-	*
		6442	*	PRIATE ERROR MESSAGE CODE (SEE ERROR_PROCEDURES).	*
		6443	*		*
		6444	*	*TABLES/WORK AREAS	*
		6445	*	N/A	*
		6446	*		*
		6447	*	*ATTRIBUTES	*
		6448	*	* REUSABLE	*
		6449	*	* RELOCATABLE	*
		6450	*		*
		6451	*	*CHARACTER CODE DEPENDENCY	*
		6452	*	THE OPERATION OF THIS MODULE DEPENDS UPON THE FOLLOWING PROPER-	*
		6453	*	TIES OF THE INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 90
6454	*		*	NUMERIC CHARACTERS 0 THROUGH 9 ARE PRESUMED TO BE CODED SUCH	*
6455	*			THAT THE HIGH ORDER FOUR BITS CONTAIN A SIGN ZONE WITH X'F'	*
6456	*			DEFINING A POSITIVE DIGIT AND X'D' DEFINING A NEGATIVE DIGIT.	*
6457	*			THE SPECIFIC INSTRUCTIONS (INSTRUCTION SEQUENCES) WHICH REQUIRE	*
6458	*			MODIFICATION IF THESE PROPERTIES OF THE CHARACTER SET ARE CHANGED	*
6459	*			MAY BE IDENTIFIED BY -	*
6460	*		*	THE 2 INSTPUCTIONS BEGINNING AT LABEL ICF005	*
6461	*		*	THE SIGN CHANGE CONSTANT AT LABEL ICRSCV	*
6462	*			COMMENTS ARE PROVIDED TO INDICATE THE CONSIDERATIONS INVOLVED AND	*
6463	*			MECHANISMS FOR CHANGING THE CODE.	*
6464	*				*
6465	*			*NOTES.	*
6466	*			ERROR PROCEDURES	*
6467	*			ICFLTA PERFORMS MOST OF ITS FUNCTIONS BY THE EXECUTION OF	*
6468	*			EXTERNAL SUBROUTINES. WHEN AN ERROR OCCURS DURING AN 'WIN-	*
6469	*			METIC FUNCTION EXECUTION, AN APPROPRIATE ERROR CODE IS LEFT IN	*
6470	*			INTERPRETER PARAMETER IZERRC. THIS PARAMETER IS TESTED WHEN	*
6471	*			CONTROL IS RETURNED TO ICFLTA AND, IF AN ERROR HAS BEEN DIS-	*
6472	*			COVERED, CONTROL IS PASSED TO THE INTERPRETER EXECUTIVE AT	*
6473	*			ENTRY POINT INTERR.	*
6474	*				*
6475	*			REGISTER USAGE	*
6476	*		*	REGISTER @BR IS EXPECTED TO CONTAIN NE INTERPRETER EXECU-	*
6477	*			TIVE. ROUTPE BASE CORE ADDRESS (IZBASE) AT ICFLTA ENTRY, AND	*
6478	*			RETAINS THIS ADDRESS AT EXIT.	*
6479	*		*	REGISTER @XR IS NOT SAVED. IT IS USED IN ICFLTA FOR GENERAL	*
6480	*			PURPOSE INDEXING.	*
6481	*				*
6482	*			SAVED/RESTORED AREAS	*
6483	*			N/A	*
6484	*				*
6485	*			MODIFICATION CONSIDERATIONS	*
6486	*			N/A	*
6487	*				*
6488	*			REQUIRED MODULES	*
6489	*		*	@SYSEQ - COMMON SYSTEM EQUATES.	*
6490	*		*	\$V\$EQU - VIRTUAL MEMORY FIXED ADDRESS EQUATES.	*
6491	*		*	\$B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.	*
6492	*		*	\$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD. PREC. ONLY).	*
6493	*		*	\$I@LEQ - INTERPRETER PARAMETER EQUATES (FOR LONG PREC. ONLY).	*
6494	*		*	FDIADD - FLOATING POINT ADD/SUBTRACT ROUTINE.	*
6495	*		*	FZIMPY - FLOATING POINT MULTIPLY ROUTINE.	*
6496	*		*	FFIDVD - FLOATING POINT DIVIDE ROUTINE.	*
6497	*		*	INTERP - INTERPRETER EXECUTIVE ROUTINE.	*
6498	*		*	IPGMDL - INTERPRETER PAGING CONTROL MODULE.	*
6499	*		*	IZCOMN - INTERPRETER COMMON ADDRESS REFERENCE EQUATES.	*
6500	*				*
6501	*			OTHER	*
6502	*			N/A	*
6503	*			*****	*

## S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	06/09/20	PAGE 91
					6505	*****				
					6506	* START OF PMC EXECUTION MODULE ICFLTA				
					6507	*****				
					6508	*				
					6509	* START ICFLTA - ESTABLISH ADDRESSABILITY				
					6510	*				
				0E32	6511	ICFLTA EQU *	START OF ICFLTA CODING			
				0C60	6512	USING IZBASE,@BR	DEFINE INTERPRETER SASE ADDRESS			
					6514	*****				
					6515	* ENTRY ICFPWR - PERFORM FLOATING POINT EXPONENTIATION				
					6516	*****				
					6517	*				
				0E32	6518	ICFPWR EQU *	ICFPWR ENTRY POINT			
0E32	5F	00	EE	E5	6519	SLC IZSTAK(,@BR),IZCL2F(@CADDR-1,@BR)	DECR THE STACK POINTER			
					6520	*				
0E36	C0	87	130B		6521	B IPGCAL	LINK TO EXECUTE POWER ROUTINE			
0E3A	0800			0E3B	6522	DC AL(@VADDR)(V\$APWR)	POWER RTN ENTRY VIRTUAL ADDR			
0E3C	F2	87	32		6523	J ICF020	BRANCH TO COMPLETE EXECUTION			
					6525	*****				
					6526	* ENTRY ICFNEG - PERFORM FLOATING POINT NEGATION				
					6527	*****				
					6528	*				
				0E3F	6529	ICFNEG EQU *	ICFNEG ENTRY POINT			
0E3F	5F	00	EE	E4	6530	SLC IZSTAK(,@BR),IZCL1F(@CADDR-1,@BR)	DECR THE STACK POINTER			
					6531	*				
0E43	75	02	EE		6532	L IZSTAK(,@BR),@XR	LOAD THE STACK POINTER			
0E46	8E	00	07	0EAC	6533	ICF005 ALC I@SIGN(,@XR),ICFSCV(1)	CHANGE THE FLOATING VALUE SIGN			
0E4B	BA	D0	07		6534	SBN I@SIGN(,@XR),B@ZNEG	* BY MANIPULATING ZONE BITS			
					6535	*				
0E4E	F2	87	26		6536	ICF007 J ICF030	BRANCH TO COMPLETE EXECUTION			
					6537	*				
					6538	*****				



S/3 BASIC INTERPRETER INITIALIZER.									
ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20 PAGE 92			
			6540	*****					
			6541	* ENTRY ICFDIV - PERFORM FLOATING POINT DIVISION					
			6542	*****					
			6543	*					
		0E51	6544	ICFDIV EQU	*	ICFDIV ENTRY POINT			
0E51	C2 02 0919		6545	LA	FFIDVD,@XR	LOAD DIVISION RTN ENTRY ADDRESS			
0E55	F2 87 12		6546	J	ICF010	BRANCH TO COMPLETE EXECUTION			
			6548	*****					
			6549	* ENTRY ICFMPY - PERFORM FLOATING POINT MULTIPLICATION					
			6550	*****					
			6551	*					
		0E58	6552	ICFMPY EQU	*	ICFMPY ENTRY POINT			
0E58	C2 02 082A		6553	LA	FZIMPY,@XR	LOAD MULTIPLY RTN ENTRY ADDRESS			
0E5C	F2 87 0B		6554	J	ICF010	BRANCH TO COMPLETE EXECUTION			
			6556	*****					
			6557	* ENTRY ICFSUB - PERFORM FLOATING POINT SUBTRACTION					
			6558	*****					
			6559	*					
		0E5F	6560	ICFSUB EQU	*	ICFSUB ENTRY POINT			
0E5F	C2 02 0751		6561	LA	FDISUB,@XR	LOAD SUBTRACT RTN ENTRY ADDRESS			
0E63	F2 87 04		6562	J	ICF010	BRANCH TO COMPLETE EXECUTION			
			6564	*****					
			6565	* ENTRY ICFADD - PERFORM-FLOATING POINT ADDITION					
			6566	*****					
			6567	*					
		0E66	6568	ICFADD EQU	*	ICFADD ENTRY POINT			
0E66	C2 02 075D		6569	LA	FDIADD,@XR	LOAD ADDITION RTN ENTRY ADDRESS			
			6570	*					
			6571	*****					

S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 93
		6573		*****	
		6574		* BINARY ARITHMETIC OPERATION EXECUTION ROUTINE	
		6575		*****	
		6576		*	
		6577		* EXECUTE ARITHMETIC OPERATION DEFINED BY THE ENTRY POINT	
		6578		*	
0E6A 5F 00 EE E5		6579	ICF010 SLC	IZSTAK(,@BR),IZCL2F(@CADDR-1,@BR) DECR THE STACK POINTER	
0E6E E0 87 00		6580	B	ICFAFN(,@XR) LINK TO PERFORM THE OPERATION	
		6581		*	
		6582		* TEST FOR AN ARITHMETIC OPERATION ERROR CONDITION	
		6583		*	
0E71 7D 00 5C		6584	ICF020 CLI	IZERRC(,@BR),I@NERR IF INTERPRETER ERROR CODE NOT	
0E74 D0 01 4B		6585	BNE	INTERR(,@BR) * NULL. GO TERMPATE ON ERROR	
		6586		*	
		6587		* COMPLETE THE PSEUDO INSTRUCTION EXECCTION	
		6588		*	
0E77 5E 00 EE E4		6589	ICF030 ALC	IZSTAK(,@BR),IZCL1F(@CADDR-1,@BR) INCR THE STACK POINTER	
0E7B D0 87 29		6590	B	INTAD1(,@BR) GO EXECUTE NEXT PSEUDO INST	
		6591		*	
		6592		*****	

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 06/09/20 PAGE 94

```

6594 *****
6595 * ENTRY ICFFN0 - PERFORM FLOATING POINT FUNCTION (NO ARGUMENT)
6596 *****
6597 *
0E7E 6598 ICFFN0 EQU * ICFFN0 ENTRY POINT
0E7E 7C 00 F7 6599 MVI IZPARM(, @BR), ICFPF0 SET 0 ARGUMENT PARAMETER
0E81 3C 04 0BA1 6600 MVI IZSLLC, I@LPFV-1 SET STACKED ARITH VALUE LENGT4
0E85 F2 87 07 6601 J ICF100 BRANCH TO COMPLETE EXECUTION

6603 *****
6604 * ENTRY ICFFNI PERFORM FLOATING POINT FUNCTION (1 ARGUMENT)
6605 *****
6606 *
0E88 6607 ICFFN1 EQU * ICFFN1 ENTRY POINT
0E88 7C 01 F7 6608 MVI IZPARM(, @BR), ICFPF1 SET 1 ARGUMENT PARAMETER
0E8B 5F 00 EE E4 6609 SLC IZSTAK(, @BR), IZCL1F(@CADDR-1, @BR) DECR THE STACK POINTER
6610 *
6611 * EXECUTE FUNCTION DEFINED BY THE PSEUDO INSTRUCTION VADDR OPERAND
6612 *
0E8F 75 02 EC 6613 ICF100 L IZXIAR(, @BR), @XR LOAD INSTRUCTION CORE ADDRESS
0E92 2C 01 0E9C 02 6614 MVC ICF110, I@XVAD(B@LCVA, @XR) MOVE INST OPERAND TO PAGE CALL
6615 * * PARAMETER AREA
0E97 C0 87 130B 6616 B IPGCAL LINK TO EXECUTE THE FUNCTION
0E9B 0E9C 6617 ICF110 DS CL(@VADDR) FUNCTION ENTRY VIRTUAL ADDRESS
6618 *
6619 * TEST FOR ARITHMETIC FUNCTION ERROR CONDITION
6620 *
0E9D 7D 00 5C 6621 ICF120 CLI IZERRC(, @BR), I@NERR IF INTERPRETER ERROR CODE NOT
0EA0 D0 01 4B 6622 BNE INTERR(, @BR) * NULL, SO TERMINATE ON ERROR
6623 *
6624 * COMPLETE THE PSELDO INSTRUCTION EXECUTION
6625 *
0EA3 7C 08 EF 6626 ICF130 MVI IZSTKI(, @BR), I@LUFV SET STACK INCREMENT
0EA6 D0 87 3D 6627 B INTADS(, @BR) LINK TO INCR THE STACK POINTER
6628 *
0EA9 D0 87 1B 6629 B INTAD3(, @BR) GO EXICUTE NUT PSEUDO INST
6630 *
6631 *****

```

S/3 BASIC INTERPRETER INITIALIZER.					
ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 95
			6633	*****	
			6634	* SCALAR ARITHMETIC ROUTINES PROGRAM CONSTANTS	
			6635	*****	
			6636	*	
0EAC 10		0EAC	6637	ICFSCV DC XL1'10'	SIGN CHANGE CONSTANT
			6639	*****	
			6640	* SCALAR ARITHMETIC ROUTINES EQUATES REFERENCING CONSTANTS	
			6641	*****	
			6642	*	
		0000	6643	ICFAFN EQU 0	DISP FOR OPERATION BRANCH ADDR
			6644	*	
		0000	6645	ICFPF0 EQU 0	0 ARGUMENT FUNCTION PARAMETER
		0001	6646	ICFPF1 EQU 1	1 ARGUMENT FUNCTION PARAMETER
			6647	*	
			6648	*****	
			6649	*	
			6650	* END OF SCALAR ARITHMETIC ROUTINES CODING	
			6651	*	

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 96
		6653		*****	
		6654	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		6655	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
		6656	*		*
		6657		*****	
		6658	*	*STATUS	*
		6659	*	VERSION 1 MODIFICATION 0	*
		6660	*		*
		6661	*	*FUNCTION	*
		6662	*	* ICMATF CONTAINS THE RUN-TIME ROUTINES WHICH INTERPRET AND CAUSE	*
		6663	*	EXECUTION OF THE FOLLOWING PSEUDO MACHINE INSTRUCTIONS -	*
		6664	*	* 'MF1' - SINGLE MATRIX FUNCTION CALL	*
		6665	*	* 'MR2' - DOUBLE MATRIX FUNCTION CALL	*
		6666	*	* 'MF3' - TRIPLE MATRI, FUNCTION CALL	*
		6667	*	* 'MSM' - MATRIX-SCALAR MULTIPLY	*
		6668	*	* THE FOLLOWING DESCRIPTIONS GIVE FUNCTIONAL SPECIFICATIONS FOR	*
		6669	*	THE ROUTINES WHICH ARE USED TO EXECUTE THE PSEUDO INSTRUCTIONS	*
		6670	*	LISTED ABOVE. EACH INSTRUCTION INVOLVES MATRIX OPERATIONS IN	*
		6671	*	THE RUN-TIME STACK.	*
		6672	*	* 'MF1' - SINGLE MATRIX FUNCTION CALL (FORMAT - OP VADR)	*
		6673	*	THE ARITHMETIC ARRAY DESCRIPTOR AT THE TOP OF THE STACK IS	*
		6674	*	USED TO DEFINE THE MATRIX ARGUMENT FOR THE FUNCTION	*
		6675	*	ROUTINE WHOSE ENTRY ADDRESS IS VADR. THE DESCRIPTOR IS	*
		6676	*	DELETED FROM THE TOP OF THE STACK AFTER FUNCTION EXECUTION.	*
		6677	*	* 'MF2' - DOUBLE MATRIX FUNCTION CALL (FORMAT - OP VADR)	*
		6678	*	THE ARITHMETIC ARRAY DESCRIPTORS AT THE SECOND AND TOP	*
		6679	*	STACK POSITIONS ARE USED TO DEFINE THE DOUBLE ARRAY ARGU-	*
		6680	*	MENTS FOR THE FUNCTION ROUTINE WHOSE ENTRY ADDRESS IS VADR.	*
		6681	*	BOTH ARRAY DESCRIPTORS ARE DELETED FROM THE STACK AFTER	*
		6682	*	FUNCTION EXECUTION.	*
		6683	*	* 'MF3' - TRIPLE MATRIX FUNCTION CALL (FORMAT - OP VADR)	*
		6684	*	THE ARITHMETIC ARRAY DESCRIPTORS AT THE THIRD, SECOND, AND	*
		6685	*	TOP STACK POSITIONS ARE USED TO DEFINE THE TRIPLE MATRIX	*
		6686	*	ARGUMENTS FOR THE FUNCTION ROUTINE WHOSE ENTRY ADDRESS IS	*
		6687	*	VADR. THE THREE ARRAY DESCRIPTORS ARE DELETED FROM THE	*
		6688	*	STACK AFTER FUNCTION EXECUTION.	*
		6689	*	* 'MSM' - MATRIX-SCALAR MULTIPLY (FORMAT - OP VADR)	*
		6690	*	THE ARITHMETIC ARRAY DESCRIPTOR AT THE THIRD STACK POSI-	*
		6691	*	TION IS USED TO DEFINE THE MATRIX. TO WHICH WILL BE	*
		6692	*	ASSIGNED THE PRODUCT ELEMENTS RESULTING FROM MULTIPLICA-	*
		6693	*	TION OF THE MATRIX DEFINED BY THE ARRAY DESCRIPTOR AT THE	*
		6694	*	TO OR THE STACK BY THE FLOATING POINT VALUE AT THE SECOND	*
		6695	*	STACK POSITION. THE ENTRY POINT OF THE FUNCTION WHICH	*
		6696	*	PERFORMS THIS OPERATION IS GIVEN BY VADR. THE MULTIPLIER	*
		6697	*	VALUE AND BOTH DESCRIPTORS ARE DELETED FROM THE STACK	*
		6698	*	AFTER FUNCTION EXECUTION.	*
		6699	*		*
		6700	*	*ENTRY POINTS	*
		6701	*	* ENTRY ICMMF1 - FOR EXECUTION OF THE 'MF1' INSTRUCTION.	*
		6702	*	CALLING SEQUENCE IS	*
		6703	*	B ICMMF1	*
		6704	*	* ENTRY ICMMF2 - FOR EXECUTION OF THE 'MF2' INSTRUCTION.	*
		6705	*	CALLING SEQUENCE IS	*
		6706	*	B ICMMF2	*
		6707	*	* ENTRY ICMMF3 - FOR EXECUTION OF THE 'MF3' INSTRUCTION.	*
		6708	*	CALLING SEQUENCE IS	*

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 97
		6709	*	B ICMMF3	*
		6710	*	* ENTRY ICMMSM - FOR EXECUTION OF THE 'MSM' INSTRUCTION,	*
		6711	*	CALLING SEQUENCE IS	*
		6712	*	B ICMMSM	*
		6713	*	* EACH OF THE ABOVE ENTRY POINTS IS ACCESSED THROUGH THE PMC	*
		6714	*	EXECUTION BRANCH ADDRESS TABLE (INTBAT) IN EXECUTIVE ROUTINE	*
		6715	*	INTERP, AND IS SUBJECT TO INPUT CONDITIONS DESCRIBED BELOW.	*
		6716	*		*
		6717	*	*INPUT	*
		6718	*	* IZSTAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER, THIS IS TO	*
		6719	*	CONTAIN THE CORE ADDRESS OF THE STACK LOCATION IMMEDIATELY	*
		6720	*	FOLLOWING THE LAST STACKED DATA ELEMENT.	*
		6721	*	* IZXIAR - 2 BYTES, FOR THE PMC ADDRESS REGISTER. THIS IS TO	*
		6722	*	CONTAIN THE CORE ADDRESS OF THE OPCODE FIELD IN THE PSEUDO	*
		6723	*	INSTRUCTION BEING EXECUTED.	*
		6724	*	* RUN-TIME STACK (FOR ENTRY POINT ICMMF1) - THIS CONTAINS AN	*
		6725	*	ARITHMETIC ARRAY DESCRIPTOR AT THE TOP STACK POSITION.	*
		6726	*	* RUN-TIME STACK (FOR ENTRY POINT ICMMF2) - THIS CONTAINS TWO	*
		6727	*	ARITHMETIC ARRAY DESCRIPTORS - ONE AT THE TOP AND ONE AT THE	*
		6728	*	SECOND STACK POSITIONS.	*
		6729	*	* RUN-TIME STACK (FOR ENTRY POINT ICMMF3) - THIS CONTAINS THREE	*
		6730	*	ARITHMETIC ARRAY DESCRIPTORS - ONE AT THE TOP, ONE AT THE	*
		6731	*	SECOND, AND ONE AT THE THIRD STACK POSITION.	*
		6732	*	* RUN-TIME STACK (FOR ENTRY POINT ICMMSM) - THIS CONTAINS TWO	*
		6733	*	ARITHMETIC ARRAY DESCRIPTORS - ONE AT THE TOP AND ONE AT THE	*
		6734	*	THIRD STACK POSITIONS. A FLOATING POINT DECIMAL VALUE IS ALSO	*
		6735	*	CONTAINED AT THE SECOND STACK POSITION.	*
		6736	*		*
		6737	*	*OUTPUT	*
		6738	*	* IZSTAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER. THIS CON-	*
		6739	*	TAINS THE CORE ADDRESS OF THE FIRST AVAILABLE STACK LOCATION	*
		6740	*	AFTER ALL INPUT ELEMENTS HAVE BEEN DELETED.	*
		6741	*	* IZERRC - 1 BYTE, FOR THE ERROR CONDITION CODE. THIS CONTAINS	*
		6742	*	A NULL CODE (I@NERR) WHEN NO ERROR CONDITION EXISTS, OR AN	*
		6743	*	ERROR CODE SPECIFYING THE PARTICULAR ERROR CONDITION DISCOVERED.	*
		6744	*	* RUN-TIME STACK - ALL INPUT ELEMENTS ARE DELETED.	*
		6745	*	* VIRTUAL MEMORY - THE AFFECTED MATRIX HAS BEEN UPDATED ACCORDING	*
		6746	*	TO THE SPECIFIED FUNCTION.	*
		6747	*		*
		6748	*	*EXTERNAL REFERENCES	*
		6749	*	* IDGCAL - ENTRY POINT FOR PAGING MODULE V.M. PROGRAM CALL RTN.	*
		6750	*	* INTAD3 - ENTRY POINT FOR INTERPRETER 3-BYTE PMC INCREMENT RTN.	*
		6751	*	* INTERR - ENTRY POINT FOR INTERPRETER EXECUTION ERROR ROUTINE.	*
		6752	*	* IZBASE - CORE ADDRESS FOR INTERP BASE ADDRESSW.ITY.	*
		6753	*	* IZSTAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER.	*
		6754	*	* IZXIAR - 2 BYTES, FOR THE PSEUDO INSTRUCTION ADDRESS REGISTER.	*
		6755	*	* IZERRC - 1 BYTE, FOR THE INTERPRETER EXECUTION ERROR CODE.	*
		6756	*	* IZCL1F - 1 BYTE, FOR LENGTH OF AN UNPACKED FLOATING POINT VALUE.	*
		6757	*		*
		6758	*	*EXITS, NORMAL	*
		6759	*	CONTROL IS NORMALLY PASSED TO THE INTERPRETER EXECUTIVE AT ENTRY	*
		6760	*	POINT INTAD3 FOR NEXT PSEUDO INSTRUCTION EXECUTION.	*
		6761	*		*
		6762	*	*EXITS, ERROR	*
		6763	*	CONTROL IS PASSED TO THE INTERPRETER EXECUTIVE AT ENTRY POINT	*
		6764	*	INTERR WITH PARAMETER IZERRC CONTAINING THE APPROPRIATE ERROR	*

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20	PAGE 98
		6765	*	MESSAGE CODE (SEE ERROR PROCEDURES),			*
		6766	*				*
		6767	*	TABEES/WORK AREAS			*
		6768	*	N/A			*
		6769	*				*
		6770	*	ATTRIBUTES			*
		6771	*	* REUSABLE			*
		6772	*	* RELOCATABLE			*
		6773	*				*
		6774	*	CHARACTER CODE DEPENDENCY			*
		6775	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR			*
		6776	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.			*
		6777	*				*
		6778	*	NOTES			*
		6779	*	ERROR PROCEDURES			*
		6780	*	ICMATF PERFORMS ITS FUNCTIONS BY THE EXECUTION OF EXTERNAL			*
		6781	*	SUBROUTINES. WHEN AN ERROR OCCURS DURING A MATRIX FUNCTION			*
		6782	*	EXECUTION, AN APPROPRIATE ERROR CODE IS LEFT IN INTERPRETER			*
		6783	*	PARAMETER IZERRC. THIS PARAMETER IS TESTED WHEN CONTROL IS			*
		6784	*	RETURNED TO ICMATF AND, IF AN ERROR HAS BEEN DISCOVERED,			*
		6785	*	CONTROL IS PASSED TO THE INTERPRETER EXECUTIVE AT ENTRY			*
		6786	*	POINT INTERRUPT.			*
		6787	*				*
		6788	*	REGISTER USAGE			*
		6789	*	* REGISTER @BR IS EXPECTED TO CONTAIN THE INTERPRETER EXECU-			*
		6790	*	TIVE ROUTINE BASE CORE ADDRESS (IZBASE) AT ICMATF ENTRY, AND			*
		6791	*	RETAINS THIS ADDRESS AT EXIT.			*
		6792	*	* REGISTER @XR IS NOT SAVED. IT IS USED IN ICMATF FOR GENERAL			*
		6793	*	PURPOSE INDEXING.			*
		6794	*				*
		6795	*	SAVED/RESTORED AREAS			*
		6796	*	N/A			*
		6797	*				*
		6798	*	MODIFICATION CONSIDERATIONS			*
		6799	*	N/A			*
		6800	*				*
		6801	*	REQUIRED MODULES			*
		6802	*	* @SYSEQ - COMMON SYSTEM EQUATES.			*
		6803	*	* \$B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.			*
		6804	*	* \$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD. PREC. ONLY).			*
		6805	*	* \$I@LEQ - INTERPRETER PARAMETER EQUATES (FOR LONG PREC. ONLY).			*
		6806	*	* INTERP - INTERPRETER EXECUTIVE ROUTINE.			*
		6807	*	* IPGMDL - INTERPRETER PAGING CONTROL MODULE.			*
		6808	*	* LZCOMN - INTERPRETER COMMON ADDRESS REFERENCE EQUATES.			*
		6809	*				*
		6810	*	OTHER			*
		6811	*	N/A			*
		6812	*	*****			*



## S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 99
			6814		*****	
			6815		* START OF PMC EXECUTION MODULE ICMATF	
			6816		*****	
			6817		*	
			6818		* START ICMATF - ESTABLISH ADDRESSABILLIV	
			6819		*	
			0EAD 6820	ICMATF EQU *	START OF ICMATF CODING	
			0C60 6821	USING IZBASE,@BR	DEFINE INTERPRETER BASE ADDRESS	
			6823		*****	
			6824		* ENTRY ICMMSM - MATRIX-SCALAR MULTIPLICATION FUNCTION	
			6825		*****	
			6826		*	
			0EAD 6827	ICMMSM EQU *	ICMMSM ENTRY POINT	
0EAD	5F	00 EE E4	6828	SLC	IZSTAK(,@BR),IZCL1F(@CADDR-1,@BR) DECR THE STACK POINTER	
0EB1	F2	87 05	6829	J	ICMMF2	BRANCH TO CONTINUE EXECUTION
			6831		*****	
			6832		* ENTRY ICMMF3 - TRIPLE MATRIX REFERENCE FUNCTION	
			6833		*****	
			6834		*	
			0EB4 6835	ICMMF3 EQU *	ICMMF3 ENTRY POINT	
0EB4	4F	00 EE 0EDA	6836	SLC	IZSTAK(,@BR),ICMLDV(@CADDR-1) DECR THE STACK POINTER	
			6838		*****	
			6839		* ENTRY ICMMF2 - DOUBLE MATRIX REFERENCE FUNCTION	
			6840		*****	
			6841		*	
			0EB9 6842	ICMMF2 EQU *	ICMMF2 ENTRY POINT	
0EB9	4F	00 EE 0EDA	6843	SLC	IZSTAK(,@BR),ICMLDV(@CADDR-1) DECR THE STACK POINTER	
			6845		*****	
			6846		* ENTRY ICMMF1 - SINGLE MATRIX REFERENCE FUNCTION	
			6847		*****	
			6848		*	
			0EBE 6849	ICMMF1 EQU *	ICMMF1 ENTRY POINT	
0EBE	4F	00 EE 0EDA	6850	SLC	IZSTAK(,@BR),ICMLDV(@CADDR-1) DECR THE STACK POINTER	
			6851		*	
			6852		*****	

## S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20 PAGE 100
				6854		*****		
				6855	*	GENERAL MATRIX FUNCTION EXECUTION ROUTINE		
				6856		*****		
				6857	*			
				6858	*	EXECUTE MATRIX FUNCTION DEFINED BY THE PSEUDO INSTRUCTION OPERAND		
				6859	*			
0EC3	75	02	EC	6860	ICM010	L	IZXIAR(,@BR),@XR	LOAD PSEUDO INSTRUCTION ADDRESS
0EC6	2C	01	0ED0 02	6861		MVC	ICM020,I@XVAD(B@LCVA,@XR)	MOVE INST OPERAND TO PAGE PARAM
				6862	*			
0ECB	C0	87	130B	6863		B	IPGCAL	LINK TO EXECUTE MATRIX FUNCTION
0ECF				6864	ICM020	DS	CL(@VADDR)	MATRIX FUNC RTN VADDR ENTRY PT
				6865	*			
				6866	*	CONTROL RETURNS WITH FUNCTION EXECUTION COMPLETED (OR ABORTED ON		
				6867	*	ERROR CONDITION) - BRANCH TO EXECUTE NEXT PSEUDO INSTRUCTION UNLESS		
				6868	*	A FUNCTION EXECUTION ERROR HAS OCCURRED		
				6869	*			
0ED1	7D	00	5C	6870	ICM030	CLI	IZERRC(,@BR),I@NERR	IF NO FUNCTION EXECUTION ERROR
0ED4	D0	81	1B	6871		BE	INTAD3(,@BR)	GO EXECUTE NEXT PSEUDO INST
				6872	*			
				6873	*	MATRIX FUNCTION ERROR CONDITION - BRANCH TO TERMINATE		
				6874	*			
0ED7	D0	87	4B	6875	ICM040	B	INTERR(,@BR)	GO TERMINATE ON MAT FUNC ERROR
				6876	*			
				6877		*****		

[illegible]

## S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20	PAGE 102
			6890		*****			
			6891	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
			6892	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
			6893	*				*
			6894		*****			*
			6895	*	STATUS			*
			6896	*	VERSION 1 MODIFICATION 0			*
			6897	*				*
			6898	*	FUNCTION			*
			6899	*	* ICELST CONTAINS THE RUN-TIME ROUTINES WHICH INTERPRET AND CAUSE			*
			6900	*	EXECUTION OF THE FOLLOWING PSEUDO MACHINE INSTRUCTIONS -			*
			6901	*	* 'STA' - STACK VIRTUAL ADDRESS			*
			6902	*	* 'STX' - STACK EXECUTION CONTROL CODE			*
			6903	*	* 'STF' - STACK FLOATING POINT VALUE			*
			6904	*	* 'USF' - UNSTACK FLOATING POINT VALUE			*
			6905	*	* 'STC' - STACK CHARACTER ELEMENT			*
			6906	*	* 'USC' - UNSTACK CHARACTER ELEMENT			*
			6907	*	* THE FOLLOWING DESCRIPTIONS GIVE FUNCTIONAL SPECIFICATIONS FOR			*
			6908	*	THE ROUTINES WHICH ARE USED TO EXECUTE THE PSEUDO INSTRUCTIONS			*
			6909	*	LISTED ABOVE. EACH INSTRUCTION INVOLVES DATA ELEMENT OPERA-			*
			6910	*	TIONS IN THE RUN-TIME STACK.			*
			6911	*	* 'STA' - STACK VIRTUAL ADDRESS (FORMAT - OP VADR)			*
			6912	*	THE VIRTUAL ADDRESS OPERAND VADR IS PLACED AT THE TOP OF			*
			6913	*	THE STACK.			*
			6914	*	* 'STX' - STACK EXECUTION CONTROL CODE (FORMAT - OP XX)			*
			6915	*	THE EXECUTION CONTROL CODE OPERAND XX IS PLACED AT THE TOP			*
			6916	*	OF THE STACK.			*
			6917	*	'STF' - STACK FLOATING POINT VALUE (FORMAT - OP VADR)			*
			6918	*	THE FLOATING POINT VALUE AT VADR IS PLACED AT THE TOP OF			*
			6919	*	THE STACK IN UNPACKED FORM.			*
			6920	*	* 'USF' - UNSTACK FLOATING POINT VALUE (FORMAT - OP)			*
			6921	*	THE FLOATING POINT VALUE AT THE TOP OF THE STACK IS STORED			*
			6922	*	IN VIRTUAL MEMORY AT THE ADDRESS CONTAINED IN THE SECOND			*
			6923	*	STACK POSITION. THE STORED VALUE AND THE REFERENCED			*
			6924	*	ADDRESS ARE DELETED FROM THE STACK.			*
			6925	*	* 'STC' - STACK CHARACTER ELEMENT (FORMAT - OP VADR)			*
			6926	*	THE CHARACTER ELEMENT AT VADR IS PLACED AT THE FOP OF THE			*
			6927	*	STACK.			*
			6928	*	* 'USC' - UNSTACK CHARACTER ELEMENT (FORMAT - OP NN)			*
			6929	*	THE CHARACTER FIELD AT THE TOP OF THE STACK IS STORED IN			*
			6930	*	VIRTUAL MEMORY AT THE ADDRESSES CONTAINED IN STACK POST-			*
			6931	*	TIONS (2) THROUGH (NN+1). THE STORED ELEMENT AND EACH OF			*
			6932	*	THE REFERENCED ADDRESSES ARE DELETED FROM THE STACK.			*
			6933	*				*
			6934	*	ENTRY POINTS			*
			6935	*	* ENTRY ICESTA - FOR EXECUTION OF THE 'STA' INSTRUCTION.			*
			6936	*	CALLING SEQUENCE IS			*
			6937	*	B ICESTA			*
			6938	*	* ENTRY ICTSTX - FOR EXECUTION OF THE 'STX' INSTRUCTION.			*
			6939	*	CALLING SEQUENCE IS			*
			6940	*	B ICSTX			*
			6941	*	* ENTRY ICESTF - FOR EXECUTION OF THE 'STF' INSTRUCTION.			*
			6942	*	CALLING SEQUENCE IS			*
			6943	*	B ICESTF			*
			6944	*	* ENTRY ICEUSF - FOR EXECUTION OF THE 'USF' INSTRUCTION.			*
			6945	*	CALLING SEQUENCE IS			*

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 103
		6946	*	B ICEUSF	*
		6947	*	* ENTRY ICTSTC - FOR EXECUTION OF THE 'STC' INSTRUCTION.	*
		6948	*	CALLING SEQUENCE IS	*
		6949	*	B ICESTC	*
		6950	*	* ENTRY ICESTF - FOR EXECUTION OF THE 'USC' INSTRUCTION.	*
		6951	*	CALLING SEQUENCE IS	*
		6952	*	B ICEUSC	*
		6953	*	* EACH OF THE ABOVE ENTRY POINTS IS ACCESSED THROUGH THE PMC	*
		6954	*	EXECUTION BRANCH ADDRESS TABLE (INTBAT) IN EXECUTION ROUTINE	*
		6955	*	INTERP, AND IS SUBJECT TO INPUT CONDITIONS DESCRIBED BELOW.	*
		6956	*		*
		6957	*	*INPUT	*
		6958	*	* IZSTAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER, THIS IS TO	*
		6959	*	CONTAIN THE CORE ADDRESS OF THE FIRST STACK LOCATION OR THE	*
		6960	*	STACK LOCATION IMMEDIATELY FOLLOWING THE LAST STACCKED DATA	*
		6961	*	ELEMENT.	*
		6962	*	* IZXIAR - 2 BYTES, FOR THE PMC ADDRESS REGISTER. THIS IS TO	*
		6963	*	CONTAIN THE CORE ADDRESS OF THE OPCODE FIELD IN THE PSEUDO	*
		6964	*	INSTRUCTION BEING EXECUTED. (THIS PARAMETER IS NOT REQUIRED	*
		6965	*	FOR ENTRY POINT ICEUSF).	*
		6966	*	* RUN-TIME STACK (FOR ENTRY POINT ICELST) - THIS CONTAINS UN-	*
		6967	*	PACKED FLOATING POINT VALUE AT THE TOP STACK POSITION. THE 2ND	*
		6968	*	STACK POSITION CONTAINS THE VIRTUAL ADDRESS OR THE DESTINATION	*
		6969	*	FIELD WHERE THIS VALEU IS TO BE STORED.	*
		6970	*	* RUN-TIME STACK (FOR ENTRY POINT ICELSC) - THIS CONTAINSS CHAR-	*
		6971	*	ACTER ELEMENT AT THE TOP STACK POSITION. THIS IS PRECEDED IN	*
		6972	*	THE STACK BY NN VIRTUAL ADDRESS ENTRIES. WHERE NN IS THE VALUE	*
		6973	*	IN THE COUNT OPERAND FIELD OF THE 'USC' INSTRUCTION. EACH	*
		6974	*	STACKED VIRTAL ADDRESS IS TO RERERENCE A DESTINATION	*
		6975	*	WHERE THE CHARACTER ELEMENT IS TO BE STORED.	*
		6976	*	* VIRTUAL MEMORY (FOR ENTRY POINTS ICESTF, ICESTC) - THIS CONTAINS*	*
		6977	*	THE SCALAR ELEMENT SPECIFIED BY THE VIRTUAL ADDRESS IN THE	*
		6978	*	INSTRUCTION OPERAND FIELD.	*
		6979	*		*
		6980	*	*OUTPUT	*
		6981	*	* IZSTAK (AFTER ENTRY POINTS ICESTA, ICESTX, ICESTF, ICESTC) -	*
		6982	*	2 BYTES, FOR THE RUN-TIME STACK POINTER. THIS CONTAINS THE	*
		6983	*	CORE ADDRESS OF THE STACK LOCATION IMMEDIATELY FOLLOWING THE	*
		6984	*	STACKED DATA ELEMENT.	*
		6985	*	* IZSTAK (AETER ENTRY POINTS ICEUSF, ICEUSC) - 2 BYTES, FOR THE	*
		6986	*	RUN-TIME STACK POINTER. THIS CONTAINS THE CORE ADDRESS OF THE	*
		6987	*	FIRST AVAILABLE STACK LOCATION AFTER ALL INPUT ELEMENTS HAVE	*
		6988	*	BEEN DELETED FROM THE STACK.	*
		6989	*	* RUN-TIME STACK (AFTER ENTRY POINTS ICESTA, ICESTX, ICESTF,	*
		6990	*	ICESTC) THIS CCNTAINS THE APPROPRIATE DATA ELEMENT AT THE TOP	*
		6991	*	STACK POSITION.	*
		6992	*	* RUN-TIME STACK (AFTER ENTRY POINTS ICEUSF, ICEUSC) - ALL DATA	*
		6993	*	ELEMENTS USED AS INPLT TO THESE ROUTINES ARE DELETED FROM THE	*
		6994	*	STACK.	*
		6995	*	* VIRTUAL MEMORY (AFTER ENTRY POINTS ICEUSF, ICEUSC) - THE UN-	*
		6996	*	STACKED DATA ELEMENT IS STORED IN THE FIELD(S) SPECIFIED BY THE	*
		6997	*	VIRTUAL ADDRESS(ES) PRECEDINS THE ELEMENT IN THE STACK (SEE	*
		6998	*	UNSTACKING ROUTINE IUSTAK).	*
		6999	*	* TRACED VARIABLE (AFTER ENTRY POINTS ICEUSF, ICEUSC) - THE UN-	*
		7000	*	STACKED DATA ELEMENT IS DISPLAYED ON THE SYSTEM PRINT DEVICE	*
		7001	*	WHEN THE OUTPUT FIELD IN VIRTUAL MEMORY IS FLAGGED FOR VARIABLE	*

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 104
		7002	*	TRACE (SEE IUSTAK).	*
		7003	*		*
		7004	*	EXTERNAL REFERENCES	*
		7005	*	* ISTACK - ENTRY POINT FOR INTERPRETER ELEMENT STACKING ROUTINE.	*
		7006	*	* IUSTAK - ENTRY POINT FOR INTERPRETER ELEMENT UNSTACKING ROUTINE.	*
		7007	*	* CUPFLT - ENTRY POINT FOR FLOATING POINT VALUE PACKING ROUTINE.	*
		7008	*	* CPUFLT - ENTRY POINT FOR FLOATING POINT VALUE UNPACKING ROUTINE.	*
		7009	*	* INTADS - ENTRY POINT FOR INTERPRETER STACK POINTER INCREMENTER.	*
		7010	*	* INTAD1 - ENTRY POINT FOR INTERPRETER 1-BYTE PMC INCREMENT RTN.	*
		7011	*	* INTAD2 - ENTRY POINT FOR INTERPRETER 2-BYTE PMC INCREMENT RTN.	*
		7012	*	* INTAD3 - ENTRY POINT FOR INTERPRETER 3-BYTE PMC INCREMENT RTN.	*
		7013	*	* IZBASE - CORE ADDRESS FOR INTERP BASE ADDRESSABILITY.	*
		7014	*	* IZSTAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER.	*
		7015	*	* IZXIAR - 2 BYTES, FOR THE PSEUDO INSTRUCTION ADDRESS REGISTER.	*
		7016	*	* IZSTKI - 1 BYTE, FOR THE STACK INCREMENT PARAMETER TO INTADS.	*
		7017	*	* IZVADR - 2 BYTES, FOR PAGING MODULE VIRTUAL ADDRESS PARAMETER.	*
		7018	*	* IZSLNG - 1 BYTE, FOR ELEMENT STACKING LENGTH PARM TO ISTACK.	*
		7019	*	* IZULNG - 1 BYTE, FOR ELEMENT UNSTACKING LENGTH PARM TO ISTACK.	*
		7020	*	* IZWRK1 - 2 BYTES, FOR INTERPRETER COMMON WORK AREA 1.	*
		7021	*	* IZCLFA - 1 BYTE, FOR LENGTH OF A VIRTUAL ADDRESS PLUS THAT OF	*
		7022	*	AN UNPACKED FLOATING POINT VALUE.	*
		7023	*	* IZCL1C - 1 BYTE, FOR LENGTH OF A CHARACTER ELEMENT.	*
		7024	*	* IZCLVA - 1 BYTE, FOR LENGTH OF A VIRTUAL ADDRESS.	*
		7025	*	* IZCBN1 - 1 BYTE, FOR INTERPRETER COMMON BINARY CONSTANT '1'.	*
		7026	*		*
		7027	*	EXITS, NORMAL	*
		7028	*	* ENTRY POINT ICEUSF - CONTROL IS PASSED TO THE INTERPRETER	*
		7029	*	EXECUTIVE AT ENTRY POINT INTAD1 FOR NEXT PSEUDO INSTRUCTION	*
		7030	*	EXECUTION.	*
		7031	*	* ENTRY POINTS ICSTX, ICEUSC - CONTROL IS PASSED TO THE INTERPRE-	*
		7032	*	TER EXECUTIVE AT ENTRY POINT INTAD2 FOR NEXT PSEUDO INSTRU-	*
		7033	*	CTION EXECUTION.	*
		7034	*	* ENTRY POINTS ICSTA, ICSTF, ICSTC - CONTROL IS PASSED TO THE	*
		7035	*	INTERPRETER EXECUTIVE AT ENTRY POINT INTAD3 OR NEXT PSEUDO	*
		7036	*	INSTRUCTION EXECUTION.	*
		7037	*		*
		7038	*	EXITS, ERROR	*
		7039	*	* ICELST UTILIZES STACK INCREMENTING ROUTINE INTADS DURING ALL	*
		7040	*	STACKING OPERATIONS. INTADS MAY ABORT EXECUTION WITHOUT RETURN-	*
		7041	*	ING CONTROL TO ICELST.	*
		7042	*		*
		7043	*	TARLES(WORK AREAS	*
		7044	*	N/A	*
		7045	*		*
		7046	*	ATTRIBUTES	*
		7047	*	* REUSABLE	*
		7048	*	* RELOCATABLE	*
		7049	*		*
		7050	*	CHARACTER CODE DEPENDENCY	*
		7051	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*
		7052	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
		7053	*		*
		7054	*	NOTES	*
		7055	*	ERROR PROCEDURES	*
		7056	*	ICELST UTILIZES INTERPRETER ROUTINE TO INCREMENT THE	*
		7057	*	RUN-TIME STACK POINTER. WHEN AN ERROR CONDITION OCCURS DURING	*

## S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20	PAGE 105
			7058	*	EXECUTION OF THIS ROUTINE, PROGRAM EXECUTION IS ABORTED BY A			*
			7059	*	DIRECT BRANCH TO INTERPRETER EXECUTIVE ENTRY POINT INTERR.			*
			7060	*				*
			7061	*	REGISTER USAGE			*
			7062	*	* REGISTER @BR IS EXPECTED TO CONTAIN THE INTERPRETER EXECU-			*
			7063	*	TIVE ROUTINE BASE CORE ADDRESS (IZBASE) AT ICELST ENTRY, AND			*
			7064	*	RETAINS THIS ADDRESS AT EXIT.			*
			7065	*	* REGISTER @XR IS NOT SAVED. IT IS USED IN ICELST FOR GENERAL			*
			7066	*	PURPOSE INDEXING.			*
			7067	*				*
			7068	*	SAVED/RESTORED AREAS			*
			7069	*	N/A			*
			7070	*				*
			7071	*	MODIFICATION CONSIDERATIONS			*
			7072	*	N/A			*
			7073	*				*
			7074	*	REQUIRED MODULES			*
			7075	*	* @SYSEQ - COMMON SYSTEM EQUATES.			*
			7076	*	* \$B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.			*
			7077	*	* \$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD. PREC. ONLY).			*
			7078	*	* \$I@LEQ - INTERPRETER PARAMETER EQUATES (FOR LONG PREC. ONLY).			*
			7079	*	* ISTACK - INTERPRETER ELEMENT STACKING ROUTINE.			*
			7080	*	* IUSTAK - INTERPRETER ELEMENT UNSTACKING ROUTINE.			*
			7081	*	* CPUFLT - FLOATING POINT VALUE UNPACKING ROUTING.			*
			7082	*	* CUPFLT - FLOATING POINT VALLE PACKING ROUTINE.			*
			7083	*	* INTERP - INTERPRETER EXECUTIVE ROUTINE.			*
			7084	*	* IPGMDL - INTERPRETER PAGING CONTROL MODULE.			*
			7085	*	* IZCOMN - INTERPRETER COMMON ADDRESS REFERENCE EQUATES.			*
			7086	*				*
			7087	*	OTHER			*
			7088	*	N/A			*
			7089	*	*****			*



## S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20	PAGE 106
					7091	*****			
					7092	* START OF PMC EXECUTION MODULE ICELST			
					7093	*****			
					7094	*			
					7095	* START ICELST - ESTABLISH ADDRESSAEILITY			
					7096	*			
				0EDB	7097	ICELST EQU *	START OF ICELST CODING		
				0C60	7098	USING IZBASE,@BR	DEFINE INTERPRETER BASE ADDRESS		
					7100	*****			
					7101	* ENTRY ICESTA - STACK VIRTUAL ADDRESS OPERAND OF AN INSTRUCTION			
					7102	*****			
					7103	*			
				0EDB	7104	ICESTA EQU *	ICESTA ENTRY POINT		
					7105	*			
					7106	* MOVE THE INSTRUCTION OPERAND TO THE RUN-TIME STACK			
					7107	*			
0EDB	75	02	EC		7108	ICE010 L IZXIAR(,@BR),@XR	LOAD INSTRUCTION CORE ADDRESS		
0EDE	75	01	EE		7109	L IZSTAK(,@BR),@BR	LOAD THE STACK POINTER		
0EE1	6C	01	01 02		7110	MVC I@SVAD(,@BR),I@XVAD(B@LCVA,@XR)	STACK THE VIRTUAL ADDR		
0EE5	C2	01	0C60		7111	LA IZBASE,@BR	RESTORE INTERPRETER BASE CDR		
					7112	*			
					7113	* COMPLFTE TNE PSEUDO INSTRUCTION EXECLITION			
					7114	*			
0EE9	7C	02	EF		7115	ICE020 MVI IZSTKI(,@BR),B@LCVA	SET STACK POINTER INCREMENT		
0EEC	D0	87	3D		7116	B INTADS(,@BR)	LINK TO INCR THE STACK POINTER		
					7117	*			
0EEF	D0	87	1B		7118	B INTAD3(,@BR)	GO EXECUTE NEXT PSEUDO INST.		
					7119	*			
					7120	*****			

S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20 PAGE 107
					7122	*****		
					7123	*		
					7124	* ENTRY ICESTX - STACK EXECUTION CODE OPERAND OR AN INSTRUCTION		
					7125	*		
		0EF2		7126	ICESTX EQU *		ICESTX ENTRY POINT	
					7127	*		
					7128	* MOVE THE INSTRUCTION OPERAND TO THE RUN-TIME STACK		
					7129	*		
	0EF2	75	02	EC	7130	ICE050 L	IZXIAR(,@BR),@XR	LOAD INSTRUCTION CORE ADDRESS
	0EF5	75	01	EE	7131		L IZSTAK(,@BR),@BR	LOAD THE STACK POINTER
	0EF8	6C	00	00 01	7132		MVC I@SCOD(,@BR),I@XCOD(B@LCXX,@XR)	STACK EXECUTION CODE
	0EFC	C2	01	0C60	7133		LA IZBASE,@BR	RESTORE INTERPRETER BASE ADDR
					7134	*		
					7135	* COMPLETE THE PSEUDO INSTRUCTION EXECUTION		
					7136	*		
	0F00	7C	01	EF	7137	ICE060 MVI	IZSTKI(,@BR),B@LCXX	SET STACK POINTER INCREMENT
	0F03	D0	87	3D	7138		B INTADS(,@BR)	LINK TO INCR THE STACK POINTER
					7139	*		
	0F06	D0	87	22	7140		B INTAD2(,@BR)	GO EXECUTE NEXT PSEUDO INST
					7141	*		
					7142	*****		

## S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20 PAGE 108
					7144	*****		
					7145	* ENTRY ICESTC - STACK A CHARACTER ELEMENT		
					7146	*****		
					7147	*		
				0F09	7148	ICESTC EQU *	ICESTC ENTRY POINT	
	0F09	3C	12	0BA2	7149	MVI IZSLNG,I@LCRV-1	SET STACK RTN LENGTH PARAMETER	
	0F0D	7C	13	EF	7150	MVI IZSTKI(,@BR),I@LCRV	SET STACK POINTER INCREMENT	
	0F10	3C	80	0F2E	7151	MVI ICE110+@Q,@NOP	SET ELEMENT UNPACKING SW OFF	
					7152	*		
	0F14	F2	87	07	7153	J ICE100	BRANCH TO CONTINUE EXECUTION	
					7155	*****		
					7156	* ENTRY ICESTF - STACK A FLOATING POINT ELEMENT		
					7157	*****		
					7158	*		
				0F17	7159	ICESTF EQU *	ICESTF ENTRY POINT	
	0F17	7C	08	EF	7160	MVI IZSTKI(,@BR),I@LUFV	SET STACK POINTER INCREMENT	
	0F1A	3C	87	0F2E	7161	MVI ICE110+@Q,@UCB	SET ELEMENT UNPACKING SW ON	
					7162	*		
					7163	* STACK ELEMENT REFERENCED BY INSTRUCTION VIRTUAL ADDRESS OPERAND		
					7164	*		
	0F1E	75	02	EC	7165	ICE100 L IZXIAR(,@BR),@XR	LOAD INSTRUCTION CORE ADDRESS	
	0F21	2C	01	144A 02	7166	MVC IZVADR,I@XVAD(B@LCVA,@XR)	SET PAGING RTN VADDR PARAMETER	
	0F26	75	02	EE	7167	L IZSTAK(,@BR),@XR	LOAD THE STACK POINTER	
	0F29	C0	87	0B50	7168	B ISTACK	LINK TO STACK THE ELEMENT	
					7169	*		
					7170	* UNPACK THE STACKED ELEMENT (FLOATING POINT ELEMENT ONLY)		
					7171	*		
	0F2D	C0	00	0A27	7172	ICE110 BC CPUFLT,*-*	LINK TO UNPACK FLT PT ELEMENT	
					7173	*		
					7174	* COMPLETE THE PSEUDO INSTRUCTION EXECUTION		
					7175	*		
	0F31	D0	87	3D	7176	ICE120 B INTADS(,@BR)	LINK TO INCR THE STACK POINTER	
					7177	*		
	0F34	D0	87	1B	7178	B INTAD3(,@BR)	GO EXECUTE NEXT PSEUDO INST.	
					7179	*		
					7180	*****		

## S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 109
			7182		*****	
			7183		* ENTRY ICEUSC - STACK A FLOATING POINT ELEMENT	
			7184		*****	
			7185		*	
		0F37	7186	ICEUSC EQU *	ICEUSC ENTRY POINT	
			7187		*	
			7188		* ESTABLISH UNSTACKING COUNT FROM THE INSTRUCTION OPERAND	
			7189		*	
0F37	75	02 EC	7190	ICE150 L	IZXIAR(, @BR), @XR LOAD INSTRUCTION CORE ADDRESS	
0F3A	6C	00 F9 01	7191	MVC	IZWRK1(, @BR), I@XCNT(B@LCNN, @XR) SET UUSTACKING COUNTER	
			7192		*	
			7193		* ESTABLISH THE CORE ADDRESS OF THE STACKED CHARACTER ELEMENT	
			7194		*	
0F3E	5F	00 EE E6	7195	ICE160 SLC	IZSTAK(, @BR), IZCL1C(@CADDR-1, @BR) DECR STACK POINTER	
0F42	1C	01 0F56 EE	7196	MVC	ICE180+@OP1, IZSTAK(@CADDR, @BR) SAVE CHAR ELEMENT CADDR	
			7197		*	
			7198		* ESTABLISH DESTINATION VADDR USING STACKED VIRTUAL ADDRESS	
			7199		*	
0F47	5F	00 EE E9	7200	ICE170 SLC	IZSTAK(, @BR), IZCLVA(@CADDR-1, @BR) DECR STACK POINTER	
0F4B	75	02 EE	7201	L	IZSTAK(, @BR), @XR LOAD THE STACK POINTER	
0F4E	2C	01 144A 01	7202	MVC	IZVADR, I@SVAD(@VADDR, @XR) SET PAGING RTN VADDR PARAMETER	
			7203		*	
			7204		* UNSTICK CHARACTER ELEMENT TO DESTINATION VIRTUAL MEMORY LOCATION	
			7205		*	
0F53	C2	02 0000	7206	ICE180 LA	*-*, @XR LOAD THE CHAR ELEMENT CADDR	
0F57	3C	12 0C3A	7207	MVI	IZULNG, I@LCRV-1 SET UNSTICK RN LENGTH PARAM	
0F5B	C0	87 0BB0	7208	B	IUSTAK LINK TO UNSTICK THE ELEMENT	
			7209		*	
			7210		* TEST FOR MORE DESTINATION VIRTUAL ADDRESSES IN THE STACK	
			7211		*	
0F5F	5F	00 F9 DE	7212	ICE190 SLC	IZWRK1(, @BR), IZCBN1(B@LCNN, @BR) DECR UNSTACKING COUNTER	
0F63	C0	84 0F47	7213	BH	ICE170 REPEAT LOOP UNTIL COUNTER = 0	
			7214		*	
0F67	D0	87 22	7215	B	INTAD2(, @BR) GO EXECUTE NEXT PSEUDO INST	
			7216		*	
			7217		*****	

S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20 PAGE 110
					7219	*****		
					7220	* ENTRY ICEUSF - UNSTACK A FLOATING POINT ELEMENT		
					7221	*****		
					7222	*		
				0F6A	7223	ICEUSF EQU *	ICEUSF ENTRY POINT	
					7224	*		
					7225	* ESTABLISH DESTINATION VADDR USING STACKED VIRTUAL ADDRESS		
					7226	*		
	0F6A	5F	00	EE	EA	7227	ICE200 SLC IZSTAK(,@BR),IZCLFA(@CADDR-1,@BR) DECR STACK POINTER	
	0F6E	75	02	EE		7228	L IZSTAK(,@BR),@XR LOAD THE STACK POINTER	
	0F71	2C	01	144A	01	7229	MVC IZVADR,I@SVAD(@VADDR,@XR) SET PAGING RTN VADDR PARM	
					7230	*		
					7231	* UNSTACK FLOATING POINT ELEMENT TO DESTINATION VIRTUAL MEMORY LOCATION		
					7232	*		
	0F76	E2	02	02	7233	ICE210 LA @VADDR(,@XR),@XR INCR STACK POINTER REGISTER		
	0F79	C0	87	0A85	7234	B CUPFLT LINK TO PACK TNE FLT PT VALUE		
	0F7D	C0	87	0BB0	7235	B IUSTAK LTNK TO UNSTACK THE ELEMENT		
					7236	*		
	0F81	D0	87	29	7237	B INTAD1(,@BR) GO EXECUTE NEXT PSEUDO INST		
					7238	*		
					7239	*****		
					7240	*		
					7241	* END OF ELEMENT STACKING PMC ROUTINES CODING		
					7242	*		

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 111
		7244		*****	
		7245	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		7246	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
		7247	*		*
		7248		*****	
		7249	*	*STATUS	*
		7250	*	VERSION 1 MODIFICATION 0	*
		7251	*		*
		7252	*	*FUNCTION	*
		7253	*	* ICARST CONTAINS THE RUN-TIME ROUTINES WHICH INTERPRET AND CAUSE	*
		7254	*	EXECUTION OF THE FOLLOWING PSEUDO MACHINE INSTRUCTIONS -	*
		7255	*	* 'SA1' - STACK VECTOR ARRAY ELEMENT ADDRESS	*
		7256	*	* 'SA2' - STACK MATRIX ARRAY ELEMENT ADDRESS	*
		7257	*	* 'SB1' - STACK CHARACTER ARRAY ELEMENT ADDRESS	*
		7258	*	* 'SF1' - STACK VECTOR ARRAY ELEMENT	*
		7259	*	* 'SF2' - STACK MATRIX ARRAY ELEMENT	*
		7260	*	* 'SC1' - STACK CHARACTER ARRAY ELEMENT	*
		7261	*	* THE FOLLOWING DESCRIPTIONS GIVE FUNCTIONAL SPECIFICATIONS FOR	*
		7262	*	THE ROUTINES WHICH ARE USED TO EXECUTE THE PSEUDO INSTRUCTIONS	*
		7263	*	LISTED ABOVE. EACH INSTRUCTION INVOLVES DATA ELEMENT OPERA-	*
		7264	*	TIONS IN THE RUN-TIME STACK.	*
		7265	*	* 'SA1' - STACK VECTOR ARRAY ELEMENT ADDR (FORMAT - OP VADR)	*
		7266	*	THE FLOATING POINT VALUE AT THE TOP OF THE STACK IS CON-	*
		7267	*	VERTED TO AN ARRAY INDEX WHICH IS USED TO DETERMINE THE	*
		7268	*	VIRTUAL ADDRESS OF AN ELEMENT IN THE 1-DIMENSIONAL ARITH-	*
		7269	*	METIC ARRAY WHOSE DESCRIPTOR IS AT VADR. THE INDEXING	*
		7270	*	VALUE IS DELETED FROM THE STACK AND THE ELEMENT VIRTUAL	*
		7271	*	ADDRESS IS PLACED AT THE TOP OF THE STACK.	*
		7272	*	* 'SA2' - STACK MATRIX ARRAY ELEMENT ADDR (FORMAT - OP VADR)	*
		7273	*	THE FLOATING POINT VALUE SECOND IN THE STACK IS CONVERTED	*
		7274	*	TO AN ARRAY ROW INDEX AND THE VALUE AT THE TOP OF THE	*
		7275	*	STACK IS CONVERTED TO AN ARRAY COLUMN INDEX, BOTH OF WHICH	*
		7276	*	ARE USED TO DETERMINE THE VIRTUAL ADDRESS OF AN ELEMENT IN	*
		7277	*	THE ARITHMETIC ARRAY WHOSE DESCRIPTOR IS AT VADR. BOTH	*
		7278	*	INDEXING VALUES ARE DELETED FROM THE STACK AND THE ELEMENT	*
		7279	*	VIRTUAL ADDRESS IS PLACED AT THE TOP OF THE STACK.	*
		7280	*	* 'SB1' - STACK CHAR. ARRAY ELEMENT ADDR (FORMAT - OP VADR)	*
		7281	*	THE FLOATING POINT VALUE AT THE TOP OF THE STACK IS CON-	*
		7282	*	VERTED TO AN ARRAY INDEX WHICH IS USED TO DETERMINE THE	*
		7283	*	VIRTUAL ADDRESS OF AN ELEMENT IN THE CHARACTER ARRAY WHOSE	*
		7284	*	DESCRIPTOR IS AT VADR. THE INDEXING VALUE IS DELETED FROM	*
		7285	*	THE STACK AND THE ELEMENT VIRTUAL ADDRESS IS PLACED AT THE	*
		7286	*	TOP OF THE STACK.	*
		7287	*	* 'SF1' - STACK VECTOR ARRAY ELEMENT (FORMAT - OP VADR)	*
		7288	*	THE FLOATING POINT VALUE AT THE TOP OF THE STACK IS CON-	*
		7289	*	VERTED TO AN ARRAY INDEX WHICH IS USED TO LOCATE AN ELE-	*
		7290	*	MENT IN THE 1-DIMENSIONAL ARITHMETIC ARRAY WHOSE DESCRIP-	*
		7291	*	TOR IS AT VADR. THE INDEXING VALUE IS DELETED FROM THE	*
		7292	*	STACK AND THE FLOATING POINT ARRAY ELEMENT IS PLACED AT	*
		7293	*	TOP OF THE STACK.	*
		7294	*	* 'SF2' - STACK MATRIX ARRAY ELEMENT (FORMAT - OP VADR)	*
		7295	*	THE FLOATING POINT VALUE SECOND IN THE STACK IS CONVERTED	*
		7296	*	TO AN ARRAY ROW INDEX AND THE VALUE AT THE TOP OF THE	*
		7297	*	STACK IS CONVERTED TO AN ARRAY COLUMN INDEX, BOTH OF WHICH	*
		7298	*	ARE USED TO LOCATE AN ELEMENT IN THE ARITHMETIC ARRAY	*
		7299	*	WHOSE DESCRIPTOR IS AT VADR. BOTH INDEXING VALUES ARE	*

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 112
		7300	*	DELETED FROM THE STACK AND THE FLOATING POINT ARRAY ELE-	*
		7301	*	MENT IS PLACED AT THE TOP OF THE STACK.	*
		7302	*	* 'SC1' - STACK CHARACTER ARRAY ELEMENT (FORMAT - OP VADR)	*
		7303	*	THE FLOATING POINT VALUE AT THE TOP OF THE STACK IS CON-	*
		7304	*	VERTED TO AN ARRAY INDEX WHICH IS USED TO LOCATE AN ELE-	*
		7305	*	MENT IN NE CHARACTER ARRAY WHOSE DESCRIPTOR IS AT VADR.	*
		7306	*	THE INDEXING VALUE IS DELETED FROM THE STACK AND THE ARRAY	*
		7307	*	CHARACTER ELEMENT IS PLACED AT THE TOP OF THE STACK.	*
		7308	*		*
		7309	*	ENTRY POINTS	*
		7310	*	* ENTRY ICASA1 - FOR EXECUTION OF THE 'SA1' INSTRUCTION.	*
		7311	*	CALLING SEQUENCE IS	*
		7312	*	B ICASA1	*
		7313	*	* ENTRY ICASA2 - FOR EXECUTION OF THE 'SA2' INSTRUCTION.	*
		7314	*	CALLING SEQUENCE IS	*
		7315	*	B ICASA2	*
		7316	*	* ENTRY ICASB1 - FOR EXECUTION OF THE 'SB1' INSTRUCTION.	*
		7317	*	CALLING SEQUENCE IS	*
		7318	*	B ICASC1	*
		7319	*	* ENTRY ICASF1 - FOR EXECUTION OF THE 'SF1' INSTRUCTION.	*
		7320	*	CALLING SEQUENCE IS	*
		7321	*	B ICASF1	*
		7322	*	* ENTRY ICASF2 - FOR EXECUTION OF THE 'SF2' INSTRUCTION.	*
		7323	*	CALLING SEQUENCE IS	*
		7324	*	B ICASF2	*
		7325	*	* ENTRY ICASC1 - FOR EXECUTION OF THE 'SC1' INSTRUCTION.	*
		7326	*	CALLING SEQUENCE IS	*
		7327	*	B ICASC1	*
		7328	*	* EACH OF THE ABOVE ENTRY POINTS IS ACCESSED THROUGH THE PMC	*
		7329	*	EXECUTION BRANCH ADDRESS TABLE (INTBAT) IN EXECUTIVE ROUTINE	*
		7330	*	INTERP, AND IS SUBJECT TO INPUT CONDITIONS DESCRIBED BELOW.	*
		7331	*		*
		7332	*	INPUT	*
		7333	*	* IZSTAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER. THIS IS TO	*
		7334	*	CONTAIN THE CORE ADDRESS OF THE STACK LOCATION FOLLOWING THE	*
		7335	*	LAST STACKED FLOATING POINT VALUE.	*
		7336	*	* IZXIAR - 2 BYTES, FOR THE PMC ADDRESS REGISTER. THIS IS TO	*
		7337	*	CONTAIN THE CORE ADDRESS OF THE OPCODE FIELD IN THE PSEUDO	*
		7338	*	INSTRUCTION BEING EXECUTED.	*
		7339	*	* RUN-TIME STACK (FOR ENTRY POINTS ICASA1, ICASB1, ICASF1,	*
		7340	*	ICASC1) - THIS CONTAINS AN UNPACKED FLOATING POINT SUBSCRIPT	*
		7341	*	VALUE AT THE TOP STACK POSITION.	*
		7342	*	* RUN-TIME STACK (FOR ENTRY POINTS ICASA2, ICASF2) - THIS CONTAINS*	*
		7343	*	TWO UNPACKED FLOATING POINT SLBSCRIPT VALUES, ONE AT THE TOP	*
		7344	*	AND ONE AT THE SECOND STACK POSITIONS.	*
		7345	*	* VIRTUAL MEMORY - THIS CONTAINS ARRAY DESCRIPTORS AND ARRAY	*
		7346	*	ELEMENTS REQUIRED DURING EXECUTION.	*
		7347	*		*
		7348	*	OUTPUT	*
		7349	*	* IZSTAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER. THIS CON-	*
		7350	*	TAINS THE CORE ADDRESS OF THE STACK LOCATION IMMEDIATELY FOL-	*
		7351	*	LOWING THE RESULTING STACKED VIRTUAL ADDRESS OR DATA ELEMENT.	*
		7352	*	* IZERRC - 1 BYTE, FOR THE ERROR CONDITION CODE. THIS CONTAINS	*
		7353	*	A NULL CODE (I@NERR) WHEN NO ERROR CONDITION EXISTS, OR AN	*
		7354	*	ERROR CODE SPECIFYING THE PARTICULAR ERROR CONDITION DISCOVERED.*	*
		7355	*	* RUN-TIME STACK - INPUT SUBSCRIPT VALUE(S) HAVE BEEN DELETED,	*



## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 113
		7356	*	AND THE STACK CONTAINS THE APPROPRIATE DATA ELEMENT AT THE TOP	*
		7357	*	STACK POSITION.	*
		7358	*		*
		7359	*	* ISTACK - ENTRY POINT FOR INTERPRETER ELEMENT STACKING ROUTINE.	*
		7360	*	CPLFLT - ENTRY POINT FOR FLOATING POINT VALUE UNPACKING ROUTINE.	*
		7361	*	CAFPBS - ENTRY POINT FOR FLT. PT. TO BINARY SUBSCRIPT CONV. RTN.	*
		7362	*	INTADS - ENTRY POINT FOR INTERPRETER STACK POINTER INCREMENTER.	*
		7363	*	INTAD3 - ENTRY POINT FOR INTERPRETER 3-BYTE PMC INCREMENT RTN.	*
		7364	*	INTERR - ENTRY POINT FOR INTERPRETER EXECUTION ERROR ROUTINE.	*
		7365	*	IZBASE - CORE ADDRESS FOR INTERP BASE ADDRESSABILITY.	*
		7366	*	IZSTAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER.	*
		7367	*	IZXIAR - 2 BYTES, FOR THE PSEUDO INSTRUCTION ADDRESS REGISTER.	*
		7368	*	IZSTKI - 1 BYTE, FOR THE STACK INCREMENT PARAMETER TO INTADS.	*
		7369	*	IZSLNG - 1 BYTE, FOR ELEMENT STACKING LENGTH PARM TO ISTACK.	*
		7370	*	IZERRC - 1 BYTE, FOR THE INTERPRETER EXECUTION ERROR CODE.	*
		7371	*	IZWRK1 - 2 BYTES, FOR INTERPRETER COMMON WORK AREA 1.	*
		7372	*	IZWRK2 - 2 BYTES, FOR INTERPRETER COMMON WORK AREA 2.	*
		7373	*	IZCL1F - 1 BYTE, FOR LENGTH OF AN UNPACKED FLOATING POINT VALUE.	*
		7374	*	IZCLVA - 1 BYTE, FOR LENGTH OF A VIRTUAL ADDRESS.	*
		7375	*	IZCBN1 - 1 BYTE, FOR INTERPRETER COMMON BINARY CONSTANT '1'.	*
		7376	*		*
		7377	*	*EXITS, NORMAL	*
		7378	*	CONTROL IS NORMALLY PASSED TO THE INTERPRETER EXECUTIVE AT ENTRY	*
		7379	*	POINT INTAD3 FOR NEXT PSEUDO INSTRUCTION EXECUTION.	*
		7380	*		*
		7381	*	*EXITS, ERROR	*
		7382	*	CONTROL IS PASSED TO THE INTERPRETER EXECUTIVE AT ENTRY POINT	*
		7383	*	INTERR WITH PARAMETER IZERRC CONTAINING THE APPROPRIATE ERROR	*
		7384	*	MESSAGE CODE (SEE ERROR PROCEDURES).	*
		7385	*		*
		7386	*	*TABLES/WORK AREAS	*
		7387	*	N/A	*
		7388	*		*
		7389	*	*ATTRIBUTES	*
		7390	*	* REUSABLE	*
		7391	*	* RELOCATABLE	*
		7392	*		*
		7393	*	*CHARACTER CODE DEPENDENCY	*
		7394	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*
		7395	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
		7396	*		*
		7397	*	*NOTES	*
		7398	*	ERROR PROCEDURES	*
		7399	*	* ERROR 1 - AN ERROR HAS OCCURRED DURING CONVERSION OF A	*
		7400	*	FLOATING POINT VALUE TO A BINARY SUBSCRIPT USING CAFPBS.	*
		7401	*	* ERROR 2 - A CONVERTED BINARY SUBSCRIPT EXCEEDS A CURRENT	*
		7402	*	DIMENSION OF THE REFERENCED ARRAY.	*
		7403	*	* IN EACH OF THESE CASES, AN ERROR CODE FCR THE MESSAGE	*
		7404	*	'SUBSCRIPT OUT OR RANGE' IS ESTABLISHED IN INTERPRETER	*
		7405	*	PARAMETER IZERRC AND CONTROL IS PASSED TO THE INTERPRETER	*
		7406	*	EXECUTIVE AT ENTRY POINT INTERR.	*
		7407	*		*
		7408	*	REGISTER USAGE	*
		7409	*	* REGISTER @BR IS EXPECTED TO CONTAIN THE INTERPRETER EXECU-	*
		7410	*	TIVE ROUTINE BASE CORE ADDRESS (IZBASE) AT ICARST ENTRY, AND	*
		7411	*	RETAINS THIS ADDRESS AT EXIT.	*

S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	06/09/20	PAGE 114
			7412	*		* REGISTER @XR IS NOT SAVED. IT IS LSED IN ICARST FOR GENERAL			*
			7413	*		PURPOSE INDEXING.			*
			7414	*					*
			7415	*	SAVED/RESTORED AREAS				*
			7416	*	N/A				*
			7417	*					*
			7418	*	MODIFICATION CONSIDERATIONS				*
			7419	*	ARRAY ELEMENT VIRTUAL ADDRESS CALCULATIONS (INSTRUCTION				*
			7420	*	SEQUENCES BEGINNING AT LABELS ICA320 AND ICA450) INVOLVE				*
			7421	*	CODING WHICH IS BASED UPON ARRAY ELEMENT LENGTH. ELEMENT				*
			7422	*	LENGTH MODIFICATION (ARITHMETIC OR CHARACTER) WILL REQUIRE				*
			7423	*	ADJUSTMENTS TO THE LOGIC INHERENT IN THIS CODING.				*
			7424	*					*
			7425	*	REQUIRED MODULES				*
			7426	*	* @SYSEQ - COMMON SYSTEM EQUATES.				*
			7427	*	* \$B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.				*
			7428	*	* \$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD. PREC. ONLY).				*
			7429	*	* \$I@LEQ - INTERPRETER PARAMETER EQUATES (FOR LONG PREC. ONLY).				*
			7430	*	* ISTACK - INTERPRETER ELEMENT STACKING ROUTINE.				*
			7431	*	* CPUFLT - FLOATING POINT VALUE UNPACKING ROUTINE.				*
			7432	*	* CAFPBS - FLOATING POINT TO BINARY SUBSCRIPT CONVERSION RTN.				*
			7433	*	* INTERP - INTERPRETER EXECUTIVE ROUTINE.				*
			7434	*	* IPGMDL - INTERPRETER PAGING CONTROL MODULE.				*
			7435	*	* IZCOMN - INTERPRETER COMMON ADDRESS REFERENCE EQUATES.				*
			7436	*					*
			7437	*	OTHER				*
			7438	*	N/A				*
			7439	*	*****				*

S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 115
				7441	*****	
				7442	* START OF PMC EXECUTION MODULE ICARST	
				7443	*****	
				7444	*	
				7445	* START ICARST - ESTABLISH ADDRESSABILITY	
				7446	*	
			0F84	7447	ICARST EQU *	START OF ICARST CODING
			0C60	7448	USING IZBASE,@BR	DEFINE INTERPRETER BASE ADDRESS
				7450	*****	
				7451	* ENTRY ICASA1 - STACK VECTOR ARRAY ELEMENT VIRTUAL ADDRESS	
				7452	*****	
				7453	*	
			0F84	7454	ICASA1 EQU *	ICASA1 ENTRY POINT
0F84	C0	87	0FD6	7455	B ICA200	LINK TO STACK THE ELEMENT VADDR
0F88	F2	87	0B	7456	J ICA010	BRANCH TO COMPLETE EXECUTION
				7458	*****	
				7459	* ENTRY ICASA2 - STACK MATRIX ARRAY ELEMENT VIRTUAL ADDRESS	
				7460	*****	
				7461	*	
			0F8B	7462	ICASA2 EQU *	ICASA2 ENTRY POINT
0F8B	C0	87	0FDD	7463	B ICA210	LINK TO STACK THE ELEMENT VADDR
0F8F	F2	87	04	7464	J ICA010	BRANCH TO COMPLETE EXECUTION
				7465	*	
				7466	*****	

S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	06/09/20	PAGE 116
					7468	*****				
					7469	* ENTRY ICASB1 - STACK CHARACTER ARRAY ELEMENT VIRTUAL ADDRESS				
					7470	*****				
					7471	*				
0F92	C0	87	1048	0F92	7472	ICASB1 EQU *	ICASB1 ENTRY POINT			
					7473	B ICA400	LINK TO STACK THE ELEMENT VADDR			
					7474	*				
					7475	* COMPLETE THE PSEUDO INSTRUCTION EXECUTION				
					7476	*				
0F96	5E	00	EE E9		7477	ICA010 ALC IZSTAK(, @BR), IZCLVA(@CADDR-1, @BR)	INCR TNE STACK POINTER			
0F9A	D0	87	1B		7478	B INTAD3(, @BR)	GO EXECUTE NEXT PSEUDO INST			
					7479	*				
					7480	*****				

## S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20 PAGE 117
					7482	*****		
					7483	* ENTRY ICASF1 - STACK VECTOR ARRAY ELEMENT FLOATING POINT VALUE		
					7484	*****		
					7485	*		
				0F9D	7486	ICASF1 EQU *	ICASF1 ENTRY POINT	
0F9D	C0	87	0FD6		7487	B ICA200	LINK TO STACK THE ELEMENT VADDR	
0FA1	F2	87	04		7488	J ICA020	BRANCH TO COMPLETE EXECUTION	
					7490	*****		
					7491	* ENTRY ICASF2 - STACK MATRIX ARRAY ELEMENT FLOATING POINT VALUE		
					7492	*****		
					7493	*		
				0FA4	7494	ICASF2 EQU *	ICASF2 ENTRY PM,	
0FA4	C0	87	0FDD		7495	B ICA210	LINK TO STACK THE ELEMENT VADDR	
					7496	*		
					7497	* STACK THE ARRAY ELEMENT FLOATING POINT VALUE		
					7498	*		
0FA8	2C	01	144A 01		7499	ICA020 MVC IZVADR,I@SVAD(@VADDR,@XR)	SET PAGING RTN VADDR PARAMETER	
0FAD	C0	87	0B50		7500	B ISTACK	LINK TO STACK THE ELEMENT	
0FB1	C0	87	0A27		7501	B CPUFLT	LINK TO UNPACK THE ELEMENT	
					7502	*		
					7503	* COMPLETE THE PSEUDO INSTRUCTION EXECUTION		
					7504	*		
0FB5	5E	00	EE E4		7505	ICA030 ALC IZSTAK(,@BR),IZCL1F(@CADDR-1,@BR)	INCR THE STACK POINTER	
0FB9	D0	87	1B		7506	B INTAD3(,@BR)	GO EXECUTE NEXT PSEUDO INST	
					7507	*		
					7508	*****		

S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20	PAGE 118
					7510	*****			
					7511	* ENTRY ICASC1 - STACK CHARACTER ARRAY ELEMENT FIELD			
					7512	*****			
					7513	*			
				0FBC	7514	ICASC1 EQU *	ICASC1 ENTRY POW		
0FBC	C0	87	1048		7515	B ICA400	LINK TO STACK THE ELEMENT VADDR		
					7516	*			
					7517	* STACK THE ARRAY ELEMENT CHARACTER FIELD			
					7518	*			
0FC0	2C	01	144A 01		7519	ICA040 MVC IZVADR,I@SVAD(@VADDR,@XR)	SET PAGING ION VADDR PARAMETER		
0FC5	3C	12	0BA2		7520	MVI IZSLNG,I@LCRV-1	SET STACK RTN LENGTN PARAMETER		
0FC9	C0	87	0B50		7521	B ISTACK	LINK TO STACK THE ELEMENT		
					7522	*			
					7523	* COMPLETE THE PSEUDO INSTRUCTION EXECUTION			
					7524	*			
0FCD	7C	13	EF		7525	ICA050 MVI IZSTKI(,@BR),I@LCRV	SET STACK POINTER INCREMENT		
0FD0	D0	87	3D		7526	B INTADS(,@BR)	LINK TO INCR THE STACK POINTER		
0FD3	D0	87	1B		7527	B INTAD3(,@BR)	GO EXECUTE NEXT PSEUDO INST		
					7528	*			
					7529	*****			

## S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 119
				7531	*****	
				7532	* ARITHMETIC ARRAY ELEMENT VIRTUAL ADDRESS STACKING ROUTINE	*
				7533	*	*
				7534	* THIS ROUTINE OPERATES ON SINGLE OR DOUBLE FLOATING POINT SUBSCRIPTS	*
				7535	* LOCATED IN THE RUN-TIME STACK. THE VIRTUAL ADDRESS OF THE ELEMENT	*
				7536	* DEFINED BY THE SUBSCRIPT(S), IN CONJUNCTION WITH THE ARRAY DESCRIP-	*
				7537	* TION LOCATED AT THE ADDRESS SPECIFIED BY THE CURRENT PSEUDO INSTRU-	*
				7538	* TION OPERAND, IS LEFT IN THE STACK IN PLACE OR THE 1ST (OR ONLY)	*
				7539	* SUBSCRIPT VALUE.	*
				7540	*	*
				7541	* INPUT -	*
				7542	* IZXIAR - CONTAINS THE CORE ADDRESS OF THE CURRENT PSEUDO INST.	*
				7543	* IZSTAK - CONTAINS THE CORE ADDRESS OF THE STACK LOCATION	*
				7544	* FOLLOWING THE SUBSCRIPT FLOATING POINT VALUE(S).	*
				7545	*	*
				7546	* OUTPUT -	*
				7547	* IZSTAK - CONTAINS THE CORE ADDRESS OF THE LEFTMOST BYTE OF THE	*
				7548	* STACKED VIRTUAL ADDRESS.	*
				7549	*****	
				7550	*	
				7551	* VECTOR ARRAY ENTRY - STACK CONTAINS A SINGLE SUBSCRIPT VALUE	
				7552	*	
0FD6 3C 87 0FFA		0FD6		7553	ICA200 EQU * VECTOR ARRAY ENTRY POINT	
0FDA F2 87 04				7554	MVI ICA250+@Q,@UCB SET SINGLE SUBSCRIPT SWITCH	
				7555	J ICA220 BRANCH TO CONTINUE PROCESSING	
				7556	*	
				7557	* MATRIX ARRAY ENTRY - STACK CONTAINS DOUBLE SUBSCRIPT VALUES	
				7558	*	
0FDD 3C 80 0FFA		0FDD		7559	ICA210 EQU * MATRIX ARRAY ENTRY POINT	
				7560	MVI ICA250+@Q,@NOP SET DOUBLE SUBSCRIPT SWITCH	
				7561	*	
				7562	* SAVE THE BRANCH ADDRESS OR RETURN TO CALLING ROUTINE	
				7563	*	
0FE1 34 08 1047				7564	ICA220 ST ICA340+@OP1,@ARR SET RETURN BRANCH ADDRESS	
				7565	*	
				7566	* ESTABLISH THE VIRTUAL ADDRESS FOR THE ARRAY DOPE VECTOR	
				7567	*	
0FE5 75 02 EC				7568	ICA230 L IZXIAR(,@BR),@XR LOAD PSEUDO INST CORE ADDRESS	
0FE8 2C 01 144A 02				7569	MVC IZVADR,I@XVAD(B@LCVA,@XR) SET PAGING RTN VADDR PARAMETER	
				7570	*	
				7571	* ESTABLISH BINARY SUBSCRIPTS FOR POSSIBLE VECTOR ARRAY	
				7572	*	
0FED 5F 01 F9 F9				7573	ICA240 SLC IZWRK1(,@BR),IZWRK1(B@LDMN,@BR) SET SUBSCRIPT-1 EQUAL 0	
0FF1 C0 87 108F				7574	B ICA600 LINK TO CONVERT SUBSCRIPT-2	
0FF5 6C 01 FB 01				7575	MVC IZWRK2(,@BR),I@SIDX(B@LDMN,@XR) SAVE BINARY SUBSCRIPT-2	
				7576	*	
				7577	* TEST FOR SINGLE OR DOUBLE STACKED SUBSCRIPT PROCESSING	
				7578	*	
0FF9 F2 00 08				7579	ICA250 JC ICA270,*-* BRANCH IF SINGLE STACKED SUBS	
				7580	*	
				7581	* ESTABLISH BINARY 'ROW' SUBSCRIPT FOR THE MATRIX ARRAY	
				7582	*	
0FFC C0 87 108F				7583	ICA260 B ICA600 LINK TO CONVERT SUBSCRIPT-1	
1000 6C 01 F9 01				7584	MVC IZWRK1(,@BR),I@SIDX(B@LDMN,@XR) SAVE BINARY SUBSCRIPT-1	
				7585	*	
				7586	* STACK THE ARITHMETIC ARRAY DOPE VECTOR	



## S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20 PAGE 120
					7587	*		
1004	3C	07	0BA2		7588	ICA270 MVI	IZSLNG,B@LADV-1	SET STACK RTN LENGTH PARAMETER
1008	C0	87	0B50		7589	B	ISTACK	LINK TO STACK THE DOPE VECTOR
					7590	*		
					7591	*	TEST FOR SUBSCRIPTS EXCEEDING CURRENT ARRAY DIMENSIONS	
					7592	*		
100C	6D	01	F9 01		7593	ICA280 CLC	IZWRK1(,@BR),B@ACD1(B@LDMN,@XR)	IF 1ST DIMENSION EXCEEDED
1010	F2	84	92		7594	JH	ICA640	* GO EXECUTE ERROR EXIT
1013	6D	01	FB 03		7595	CLC	IZWRK2(,@BR),B@ACD2(B@LDMN,@XR)	IF 2ND DIMENSION EXCEEDED
1017	F2	84	8B		7596	JH	ICA640	* GO EXECUTE ERROR EXIT
					7597	*		
					7598	*	COMPUTE THE ARRAY INDEX (SUB2*(SUB1-1)*DIM2)	IN WORK AREA 2
					7599	*		
101A	F2	87	04		7600	ICA290 J	ICA310	BRANCH TO INITIALLY DECR SUB1
					7601	*		
101D	6E	01	FB 03		7602	ICA300 ALC	IZWRK2(,@BR),B@ACD2(B@LDMN,@XR)	ADD DIM2 TO WORK AREA
1021	5F	01	F9 DE		7603	ICA310 SLC	IZWRK1(,@BR),IZCBN1(B@LDMN,@BR)	DECREMENT SUBSCRIPT-1
1025	C0	84	101D		7604	BH	ICA300	REPEAT MPY LOOP UNTIL SUB1 = 0
					7605	*		
					7606	*	DETERMINE THE ARRAY ELEMENT VIRTUAL ADDRESS - THE FOLLOWING CODING	
					7607	*	IS BASED ON PACKED FLOATING POINT DATA LENGTHS OF 5 AND 9 BYTES FOR	
					7608	*	SHORT AND LONG PRECISION RESPECTIVELY.	
					7609	*		
1029	9C	01	01 FB		7610	ICA320 MVC	I@SVAD(,@XR),IZWRK2(@VADDR,@BR)	SET VADDR = 1 * INDEX
102D	5E	01	FB FB		7611	ALC	IZWRK2(,@BR),IZWRK2(@VADDR,@BR)	CALC 2 * ARRAY INDEX
1031	5E	01	FB FB		7612	ALC	IZWRK2(,@BR),IZWRK2(@VADDR,@BR)	CALC 4 * ARRAY INDEX
					7613	*		
1035	F2	87	04		7614	JC	ICA330,I@PRSW	BRANCH IF STANDARD PREC
					7615	*		
1038	5E	01	FB FB		7616	ALC	IZWRK2(,@BR),IZWRK2(@VADDR,@BR)	CALC 8 * ARRAY INDEX
					7617	*		
103C	9E	01	01 FB		7618	ICA330 ALC	I@SVAD(,@XR),IZWRK2(@VADDR,@BR)	VADDR = LENGTH * INDEX
1040	AE	01	01 07		7619	ALC	I@SVAD(,@XR),B@ABAS(@VADDR,@XR)	ADD THE ARRAY BASE VADDR
					7620	*		
					7621	*	RETURN CONTROL TO THE CALLING ICARST ROUTINE	
					7622	*		
1044	C0	87	0000		7623	ICA340 B	*-*	RETURN TO CALLING ROUTINE
					7624	*****		

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 06/09/20 PAGE 121

```

7626 *****
7627 * CHARACTER ARRAY ELEMENT VIRTUAL ADDRESS STACKING ROUTINE *
7628 * *
7629 * THIS ROUTINE OPERATES ON A SINGLE FLOATING POINT SUBSCRIPT LOCATED *
7630 * IN THE RUN-TIME STACK. THE VIRTUAL ADDRESS OF THE ELEMENT DEFINED *
7631 * BY THE SUBSCRIPT, IN CONJUNCTION WITH THE ARRAY DESCRIPTION LOCATED *
7632 * AT THE ADDRESS SPECIFIED BY THE CURRENT PSEUDO INSTRUCTION OPERAND, *
7633 * IS LEFT IN THE STACK IN PLACE OR THE SUBSCRIPT VALUE. *
7634 * *
7635 * INPUT - *
7636 * IZXIAR - CONTAINS THE CORE ADDRESS OF THE CURRENT PSEUDO INST. *
7637 * IZSTAK - CONTAINS THE CORE ADDRESS OF THE STACK LOCATION FOLLOW- *
7638 * ING THE SUBSCRIPT FLOATING POINT VALUE. *
7639 * *
7640 * OUTPUT - *
7641 * IZSTAK - CONTAINS THE CORE ADDRESS OF THE LEFTMOST BYTE OF THE *
7642 * STACKED VIRTUAL ADDRESS. *
7643 *****
7644 *
1048 34 08 108E 1048 7645 ICA400 EQU * CHARACTER ARRAY ENTRY POINT
7646 ST ICA460+@OP1,@ARR SET RETUR% BRANCH ADDRESS
7647 *
7648 * ESTABLISH THE VIRTUAL ADDRESS FOR THE ARRAY DOPE VECTOR
7649 *
104C 75 02 EC 7650 ICA410 L IZXIAR(,@BR),@XR LOAD PSEUDO INST CORE ADDRESS
104F 2C 01 144A 02 7651 MVC IZVADR,I@XVAD(B@LCVA,@XR) SET PAGING RT VADDR PARAMETER
7652 *
7653 * ESTABLISH BINARY SUBSCRIPT FOR THE CHARACTER ARRAY
7654 *
1054 C0 87 108F 7655 ICA420 B ICA600 LINK TO CONVERT THE SUBSCRIPT
1058 6C 01 F9 01 7656 MVC IZWRK1(,@BR),I@SIDX(B@LDMN,@XR) SAVE THE BINARY SUBSCRIPT
7657 *
7658 * STACK THE CHARACTER ARRAY DOPE VECTOR
7659 *
105C 3C 03 0BA2 7660 ICA430 MVI IZSLNG,B@LCDV-1 SET STACK RTN LENGTH PARAMETER
1060 C0 87 0B50 7661 B ISTACK LINK TO STACK THE DOPE VECTOR
7662 *
7663 * TEST SUBSCRIPT EXCEEDING THE CHARACTER ARRAY DIMENSION
7664 *
1064 6D 01 F9 01 7665 ICA440 CLC IZWRK1(,@BR),B@CDMN(B@LDMN,@XR) IF DIMENSION IS EXCEEDED
1068 F2 84 3A 7666 JH ICA640 * GO EXECUTE ERROR EXIT
7667 *
7668 * DETERMINE THE ARRAY ELEMENT VIRTUAL ADDRESS - THE FOLLOWING CODING
7669 * IS BASED ON A CHARACTER ELEMENT DATA LENGTH OF 19 BYTES, WORK AREA
7670 * CONTAINS THE ARRAY INOP:.
7671 *
106B 9C 01 01 F9 7672 ICA450 MVC I@SVAD(,@XR),IZWRK1(@VADDR,@BR) SET VADDR = 1 * INDEX
106F 5E 01 F9 F9 7673 ALC IZWRK1(,@BR),IZWRK1(@VADDR,@BR) CALC 2 * ARRAY INDEX
1073 9E 01 01 F9 7674 ALC I@SVAD(,@XR),IZWRK1(@VADDR,@BR) VADDR = 3 * ARRAY INDEX
1077 5E 01 F9 F9 7675 ALC IZWRK1(,@BR),IZWRK1(@VADDR,@BR) CALC 4 * ARRAY INDEX
107B 5E 01 F9 F9 7676 ALC IZWRK1(,@BR),IZWRK1(@VADDR,@BR) CALC 8 * ARRAY INDEX
107F 5E 01 F9 F9 7677 ALC IZWRK1(,@BR),IZWRK1(@VADDR,@BR) CALC 16 * ARRAY INDEX
1083 9E 01 01 F9 7678 ALC I@SVAD(,@XR),IZWRK1(@VADDR,@BR) VADDR = LENGTH * INDEX
1087 AE 01 01 03 7679 ALC I@SVAD(,@XR),B@CBAS(@VADDR,@XR) ADD THE ARRAY BASE VADDR
7680 *
7681 * RETURN CONTROL TO THE CALLING ICARST ROUTINE

```

[illegible]

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 123
		7687		*****	
		7688	*	STACKED SUBSCRIPT FLOATING POINT TO BINARY CONVERSION ROUTINE	*
		7689	*		*
		7690	*	THIS ROUTINE OPERATES ON A FLOATING POINT SUBSCRIPT LOCATED IN THE	*
		7691	*	RUN-TIME STACK. THE EQUIVALENT BINARY SUBSCRIPT IS LEFT IN THE	*
		7692	*	STACK IN PLACE OF THE FLOATING POINT VALUE.	*
		7693	*		*
		7694	*	INPUT -	*
		7695	*	IZSTAK - CONTAINS THE CORE ADDRESS OF THE STACK LOCATION FOLLOW-	*
		7696	*	ING THE SUBSCRIPT FLOATING POINT VALUE.	*
		7697	*	OUTPUT -	*
		7698	*	IZSTAK - CONTAINS TNE CORE ADDRESS OF THE LEFT BYTE OF THE	*
		7699	*	STACKED BINARY SUBSCRIPT.	*
		7700		*****	
		7701	*		
108F 34 08 10A4		108F 7702	ICA600 EQU *	CONVERSION ROUTINE ENTRY PONT	
		7703	ST ICA630+@OP1,@ARR	SET RETURN BRANCH ADDRESS	
		7704	*		
		7705	*	CONVERT THE FLOATING POINT SUBSCRIPT TO BINARY	
		7706	*		
1093 5F 00 EE E4		7707	ICA610 SLC IZSTAK(,@BR),IZCL1F(@CADDR-1,@BR)	DECR THE STACK POINTER	
1097 75 02 EE		7708	L IZSTAK(,@BR),@XR	LOAD THE STACK POINTER	
109A C0 87 0AE3		7709	B CAFPBS	LINK TO CONVERT THE SUBSCRIPT	
		7710	*		
		7711	*	TEST FOR SUCCESSFUL CONVERSION - RETURN IF NO ERROR FOUND	
		7712	*		
109E 7D 00 5C		7713	ICA620 CLI IZERRC(,@BR),I@NERR	IF NO CONVERSION ERROR	
10A1 C0 81 0000		7714	ICA630 BE *-*	* RETURN TO CALLING ROUTINE	
		7715	*		
		7716	*	ERROR EXIT - SET ERROR ROUTINE TO DISPLAY 'SUBSCRIPT OUT OF RANGE'	
		7717	*		
10A5 7C D1 5C		7718	ICA640 MVI IZERRC(,@BR),@@E760	SET INC ERROR MESSAGE CODE	
10A8 D0 87 4B		7719	B INTERR(,@BR)	GO TERMINATE ON SUBSCRIPT ERROR	
		7720	*		
		7721		*****	
		7722	*		
		7723	*	END OF ARRAY ELEMENT STACKING PMC ROUTINES CODING	
		7724	*		

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20	PAGE 124
		7726		*****			
		7727	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
		7728	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
		7729	*				*
		7730		*****			
		7731	*	STATUS			*
		7732	*	VERSION 1 MODIFICATION 0			*
		7733	*				*
		7734	*	FUNCTION			*
		7735	*	* ICTEST CONTAINS THE RUN-TIME ROUTINES WHICH INTERPRET AND CAUSE			*
		7736	*	EXECUTION OF THE FOLLOWING PSEUDO MACHINE INSTRUCTIONS -			*
		7737	*	* 'CMF' - COMPARE FLOATING POINT VALUES			*
		7738	*	* 'CMC' - COMPARE CHARACTER ELEMENTS			*
		7739	*	* 'CSA' - COMPUTE STACKED ADDRESS			*
		7740	*	* THE FOLLOWING DESCRIPTIONS GIVE FUNCTIONAL SPECIFICATIONS FOR			*
		7741	*	THE ROUTINES WHICH ARE USED TO EXECUTE THE PSEUDO INSTRUCTIONS			*
		7742	*	LISTED ABOVE, EACH INSTRUCTION INVOLVES DATA ELEMENT OPERA-			*
		7743	*	TIONS IN THE RUN-TIME STACK.			*
		7744	*	* 'CMF' - COMPARE FLOATING POINT VALUES (FORMAT - OP)			*
		7745	*	THE FLOATING POINT VALUE SECOND IN THE STACK IS COMPARED			*
		7746	*	ALGEBRAICALLY TO THE FLOATING POINT VALUE AT THE TOP OF			*
		7747	*	THE STACK. A COMPARE CONDITION CODE IS SET SPECIFYING			*
		7748	*	GREATER THAN, EQUAL TO, OR LESS THAN, AND BOTH OF THE			*
		7749	*	VALUES ARE DELETED FROM THE STACK.			*
		7750	*	* 'CMC' - COMPARE CHARACTER ELEMENTS (FORMAT - OP)			*
		7751	*	THE CHARACTER FIELD SECOND IN THE STACK IS COMPARED WITH			*
		7752	*	THE CHARACTER FIELD AT THE TOP OF THE STACK. A COMPARE			*
		7753	*	CONDITION CODE IS SET SPECIFYING A COLLATING SEQUENCE			*
		7754	*	GREATER THAN, EQUAL TO, OR LESS THAN, AND BOTH OF THE			*
		7755	*	CHARACTER ELEMENTS ARE DELETED FROM THE STACK.			*
		7756	*	* 'CSA' - COMPUTE STACKED ADDRESS (FORMAT - OP NN)			*
		7757	*	THE FLOATING POINT VALUE AT THE TOP OF THE STACK IS CON-			*
		7758	*	VERTED TO AN INDEX WHICH IS USED TO REFERENCE ONE OF THE			*
		7759	*	VIRTUAL ADDRESSES IN THE SERIES OF ADDRESSES CONTAINED IN			*
		7760	*	STACK POSITIONS (NN+1) THROUGH (2). AN INDEX VALUE LESS			*
		7761	*	THAN 1 OR GREATER THAN NN CAUSES THE VIRTUAL ADDRESS AT			*
		7762	*	STACK POSITION (NN+2) TO BE REFERENCED INSTEAD. THE			*
		7763	*	INDEXING VALUE AND THE SERIES OF ADDRESSES IN POSITIONS			*
		7764	*	(NN+2) THROUGH (2) ARE DELETED FROM THE STACK AND THE			*
		7765	*	REFERENCED VIRTUAL ADDRESS IS PLACED AT THE TOP OF THE			*
		7766	*	STACK.			*
		7767	*				*
		7768	*	ENTRY POINTS			*
		7769	*	* ENTRY ICTCMF - FOR EXECUTION OF THE 'CMF' INSTRUCTION.			*
		7770	*	CALLING SEQUENCE IS			*
		7771	*	B ICTCMF			*
		7772	*	* ENTRY ICTCMC - FOR EXECUTION OF THE 'CMC' INSTRUCTION.			*
		7773	*	CALLING SEQUENCE IS			*
		7774	*	B ICTCMC			*
		7775	*	* ENTRY ICTCSA - FOR EXECUTION OF THE 'CSA' INSTRUCTION.			*
		7776	*	CALLING SEQUENCE IS			*
		7777	*	B ICTCSA			*
		7778	*	* EACH OF THE ABOVE ENTRY POINTS IS ACCESSED THROUGH THE PMC			*
		7779	*	EXECUTION BRANCH ADDRESS TABLE (INTBAT) IN EXECUTION ROUTINE			*
		7780	*	INTERP, AND IS SUBJECT TO INPUT CONDITIONS DESCRIBED BELOW.			*
		7781	*				*

## S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 125
			7782	*	INPUT	*
			7783	*	* IZSTAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER. THIS IS TO	*
			7784	*	CONTAIN THE CORE ADDRESS OF THE STACK LOCATION FOLLOWING THE	*
			7785	*	LAST STACKED DATA ELEMENT.	*
			7786	*	* RUN-TIME STACK (FOR ENTRY POINT ICTCMF) - THIS CONTAINS TO	*
			7787	*	UNPACKED FLOATING POINT VALUES - ONE AT THE TOP AND ONE AT THE	*
			7788	*	SECOND STACK POSITIONS.	*
			7789	*	* RUN-TIME STACK (FOR ENTRY POINT ICTCMC) - THIS CONTAINS TWO	*
			7790	*	CHARACTER ELEMENTS - ONE AT THE TOP AND ONE AT THE SECOND STACK	*
			7791	*	POSITION.	*
			7792	*	* RUN-TIME STACK (FOR ENTRY POINT ICTCSA) - THIS CONTAINS AN UN-	*
			7793	*	PACKED FLOATING POINT INDEX VALUE AT THE TOP STACK POSITION.	*
			7794	*	THIS IS PRECEDED IN THE STACK BY (NN+1) VIRTUAL ADDRESS ENTRIES,	*
			7795	*	WHERE NN IS THE VALUE IN THE COUNT OPERAND FIELD OF THE 'CSA'	*
			7796	*	PSEUDO INSTRUCTION.	*
			7797	*	* IZXIAR (FOR ENTRY POINT ICTCSA) - 2 BYTES, FOR THE PMC ADDRESS	*
			7798	*	REGISTER. THIS IS TO CONTAIN THE CORE ADDRESS OF THE OPCODE	*
			7799	*	FIELD IN THE PSEUDO INSTRUCTION BEING EXECUTED.	*
			7800	*		*
			7801	*	OUTPUT	*
			7802	*	* IZBRN (AFTER ENTRY POINTS ICTCMF, ICTCMC) - 1 BYTE, FOR THE	*
			7803	*	INTERPRETER COMPARE CONDITION CODE. THIS IS GET TO ONE OF THE	*
			7804	*	FOLLOWING CODES DEPENDING ON THE RESULT OF THE COMPARE	*
			7805	*	OPERATION -	*
			7806	*	* LOW COMPARE - X'02'	*
			7807	*	* EQUAL COMPARE - X'04'	*
			7808	*	* HIGH COMPARE - X'08'	*
			7809	*	* IZSTAK (AFTER ENTRY POINTS ICTCMF, ICTCMC) - 2 BYTES, FOR THE	*
			7810	*	RUN-TIME STACK POINTER. THIS CONTAINS THE CORE ADDRESS OF THE	*
			7811	*	FIRST AVAILABLE STACK LOCATION AFTER BOTH INPUT ELEMENTS HAVE	*
			7812	*	BEEN DELETED.	*
			7813	*	* IZSTAK (AFTER ENTRY POINT ICTCSA) - 2 BYTES, FOR THE RUN-TIME	*
			7814	*	STACK POINTER. THIS CONTAINS THE CORE ADDRESS OF A THE STACK	*
			7815	*	LOCATION IMMEDIATELY FOLLOWING THE VIRTUAL ADDRESS STACKED AS A	*
			7816	*	RESULT OF THE 'CSA' OPERATION (SEE FUNCTION).	*
			7817	*	* RUN-TIME STACK (AFTER ENTRY POINTS ICTCMF, ICTCMC) - BOTH INPUT	*
			7818	*	ELEMENTS HAVE BEEN DELETED FROM THE STACK.	*
			7819	*	* RUN-TIME STACK (AFTER ENTRY POINT ICTCSA) - ALL INPUT ELEMENTS	*
			7820	*	ARE DELETED, AND THE STACK CONTAINS THE RESULTING VIRTUAL	*
			7821	*	ADDRESS IN THE TOP STACK POSITION.	*
			7822	*		*
			7823	*	EXTERNAL REFERENCES	*
			7824	*	* CAFPBS - ENTRY POINT FOR FLT. PT. TO BINARY INDEX CONV. ROUTINE.	*
			7825	*	* INTAD1 - ENTRY POINT FOR INTERPRETER 1-BYTE PMC INCREMENT RTN.	*
			7826	*	* INTAD2 - ENTRY POINT FOR INTERPRETER 2-BYTE PMC INCREMENT RTN.	*
			7827	*	* IZBASE - CORE ADDRESS FOR INTERP BASE ADDRESSABILITY.	*
			7828	*	* IZSTAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER.	*
			7829	*	* IZXIAR - 2 BYTES, FOR THE PSEUDO INSTRUCTION ADDRESS REGISTER.	*
			7830	*	* IZBRN - 1 BYTE, FOR THE ELEMENT COMPARE CONDITION CODE.	*
			7831	*	* IZERRC - 1 BYTE, FOR THE INTERPRETER EXECUTION ERROR CODE.	*
			7832	*	* IZWRK1 - 2 BYTES, FOR INTERPRETER COMMON WORK AREA 1.	*
			7833	*	* IZCL1F - 1 BYTE, FOR LENGTH OF AN UNPACKED FLOATING POINT VALUE.	*
			7834	*	* IZCL2F - 1 BYTE, FOR LENGTH OF 2 UNPACKED FLOATING POINT VALUES.	*
			7835	*	* IZCL2C - 1 BYTE, FOR LENGTH OF 2 CHARACTER ELEMENTS.	*
			7836	*	* IZCBM1 - 2 BYTES, FOR INTERPRETER COMMON BINARY CONSTANT '-1'.	*
			7837	*		*



## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 126
		7838	*	EXITS, NORMAL	*
		7839	*	* ENTRY POINTS ICTCMF, ICTCMC - CONTROL IS PASSED TO THE INTER-	*
		7840	*	PRETER EXECUTIVE AT ENTRY POINT INTAD1 FOR NEXT PSEUDO INSTRU-	*
		7841	*	TION EXECUTION.	*
		7842	*	* ENTRY POINT ICTCSA - CONTROL IS PASSED TO THE INTERPRETER	*
		7843	*	EXECUTIVE AL ENTRY POINT INTAD2 FOR NEXT PSEUDO INSTRUCTION	*
		7844	*	EXECUTION.	*
		7845	*		*
		7846	*	ERROR EXITS	*
		7847	*	N/A	*
		7848	*		*
		7849	*	TABLES/WORK AREAS	*
		7850	*	N/A	*
		7851	*		*
		7852	*	ATTRIBUTES	*
		7853	*	* REUSABLE	*
		7854	*	* RELOCATABLE	*
		7855	*		*
		7856	*	CHARACTER CODE DEPENDENCY	*
		7857	*	THE OPERATION OF THIS MODULE DEPENDS UPON THE FOLLOWING PROPER-	*
		7858	*	TIES OF THE INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
		7859	*	* NUMERIC CHARACTERS 0 THROUGH 9 ARE PRESUMED TO BE CODED SUCH	*
		7860	*	THAT THE HIGH ORDER FOUR BITS CONTAIN A SIGN ZONE WITH 'F'	*
		7861	*	DEFINING A POSITIVE DIGIT.	*
		7862	*	THE SPECIFIC INSTRUCTIONS (INSTRUCTION SEQUENCES) WHICH REQUIRE	*
		7863	*	MODIFICATION IF THESE PROPERTIES OF THE CHARACTER SET ARE CHANGED	*
		7864	*	MAY BE IDENTIFIED BY -	*
		7865	*	* THE 4 INSTRUCTIONS BEGINNING AT LABEL ICT032	*
		7866	*	* THE 2 INSTRUCTIONS BEGINNING AT LABEL ICT035	*
		7867	*	COMMENTS ARE PROVIDED TO INDICATE THE CONSIDERATIONS INVOLVED AND	*
		7868	*	MECHANISMS FOR CHANGING THE CODE.	*
		7869	*		*
		7870	*	NOTES	*
		7871	*	ERROR PROCEDURES	*
		7872	*	N/A	*
		7873	*		*
		7874	*	REGISTER USAGE	*
		7875	*	* REGISTER @BR IS EXPECTED TO CONTAIN THE INTERPRETER EXEC-	*
		7876	*	TIVE ROUTINE BASE CORE ADDRESS (IZBASE) AT ICIEST ENTRY, AND	*
		7877	*	RETAINS THIS ADDRESS AT EXIT.	*
		7878	*	* REGISTER @XR IS NOT SAVED. IT IS USED IN ICTEST FOR GENERAL	*
		7879	*	PURPOSE INDEXING.	*
		7880	*		*
		7881	*	SAVED/RESTORED AREAS	*
		7882	*	N/A	*
		7883	*		*
		7884	*	MODIFICATION CONSIDERATIONS	*
		7885	*	N/A	*
		7886	*		*
		7887	*	REQUIRED MODULES	*
		7888	*	* @SYSEQ - COMMON SYSTEM EQUATES.	*
		7889	*	* \$B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.	*
		7890	*	* \$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD. PREC. ONLY).	*
		7891	*	* \$I@LEQ - INTERPRETER PARAMETER EQUATES (FOR LONG PREC. ONLY).	*
		7892	*	* CAFPBS - FLOATING POINT TO BINARY INDEX CONVERSION ROUTINE.	*
		7893	*	* INTERP - INTERPRETER EXECUTIVE ROUTINE.	*



S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	06/09/20	PAGE 127
				7894	*		* ICBRAN - INTERPRETER BRANCH PMC EXECUTION ROUTINE.			*
				7895	*		* IZCOMN - INTERPRETER COMMON ADDRESS REFERENCE EQUATES.			*
				7896	*					*
				7897	*	OTHER				*
				7898	*	N/A				*
				7899	*****					

## S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	06/09/20	PAGE 128
					7901	*****				
					7902	* START OF PMC EXECUTION MODULE ICTEST				
					7903	*****				
					7904	*				
					7905	* START ICTEST - ESTABLISH ADRESSABILITY				
					7906	*				
				10AB	7907	ICTEST EQU *	START OF ICTEST CODING			
				0C60	7908	USING IZBASE,@BR	DEFINE INTERPRETER BASE ADDRESS			
					7910	*****				
					7911	* ENTRY ICTCMC - COMPARE CHARACTER ELEMENTS				
					7912	*****				
					7913	*				
				10AB	7914	ICTCMC EQU *	ICTCMC ENTRY POINT			
					7915	*				
					7916	* COMPARE STACK ELEMENT-2 TO STACK ELEMENT-1 (CHARACTER ELEMENTS)				
					7917	*				
10AB	5F	00	EE	E7	7918	ICT010 SLC IZSTAK(,@BR),IZCL2C(@VADDR-1,@BR)	DECR PT TO 2ND ELEMENT			
10AF	75	02	EE		7919	L IZSTAK(,@BR),@XR	LOAD STACK POINTER			
10B2	AD	11	12	25	7920	CLC I@LCRF(,@XR),I@LCRV+I@LCRF(I@LCRF,@XR)	COMPARE E2 TO E1			
					7921	*	* WHERE E1 IS TOP STACK ELEMENT			
					7922	*				
					7923	* BRANCH TO SET LOW, EQUAL OR HIGH COMPARE CONDITION STATUS				
					7924	*				
10B6	F2	82	34		7925	ICT020 JL ICT050	BRANCH IF 2ND ELEM < 1ST ELEM			
10B9	F2	81	38		7926	JE ICT060	BRANCH IF 2ND ELEM = 1ST ELEM			
10BC	F2	84	3C		7927	JH ICT070	BRANCH IF 2ND ELEM > 1ST ELEM			
					7928	*				
					7929	*****				

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15,	MOD 00	06/09/20	PAGE 129
					7931		*****				
					7932	*	ENTRY ICTCPT - COMPARE FLOATING POINT ELEMENTS				
					7933		*****				
					7934	*					
				10BF	7935	ICTCMF EQU	*				
					7936	*					
					7937	*	TEST SIGNS OR ELEMENTS TO BE COMPARED - SET APPROPRIATE COMPARE				
					7938	*	CONDITION CODES WHEN SIGNS ARE DIFFERENT IN A STATEMENT OF THE				
					7939	*	FORM: IF E1 (RELATIONAL OPERATOR) E2 THEN NNN				
					7940	*					
	10BF	5F	00	EE	E5	7941	ICT030 SLC	IZSTAK(,@BR),IZCL2F(@VADDR-1,@BR)	DECR PT TO 2ND ELEMENT		
	10C3	75	02	EE		7942	L	IZSTAK(,@BR),@XR	LOAD THE STACK POINTER		
					7943	*					
	10C6	B8	F0	07		7944	ICT032 TBN	I@1SE1+I@SIGN(,@XR),B@ZPOS	IF E1 IS POSITIVE VALUE		
	10C9	F2	10	0D		7945	JT	ICT035	* SO TEST SISN OF ELEMENT E2		
	10CC	B8	F0	0F		7946	TBN	I@1SE2+I@SIGN(,@XR),B@ZPOS	IF E2 IS POSITIVE VALUE		
	10CF	F2	10	1B		7947	JT	ICT050	* GO SET LOW COMPARE CONDITION		
	10D2	3C	82	10EB		7948	MVI	ICT045+@Q,@BL	* ELSE SET BRANCH FOR 2 - VALS		
	10D6	F2	87	0A		7949	J	ICT040	* AND SKIP TO PROCESS LIKE SGNS		
	10D9	B8	F0	0F		7950	ICT035 TBN	I@1SE2+I@SIGN(,@XR),B@ZPOS	IF E2 IS NOT POSITIVE VALUE		
	10DC	F2	90	1C		7951	JF	ICT070	* SO SET HIGH COMPARE CONDITION		
	10DF	3C	84	10EB		7952	MVI	ICT045+@Q,@BH	* ELSE SET BRANCH FOR 2 + VALS		
					7953	*					
					7954	*	ELEMENT SIGNS IDENTICAL - COMPARE STACK ELEMENT-2 TO STACK ELEMENT-1				
					7955	*	AND SET LOW, EQUAL, CR SIGH CONDITION CODE DEPENDING ON RESULT				
					7956	*					
	10E3	AD	07	07	0F	7957	ICT040 CLC	I@RSE1(,@XR),I@RSE2(I@LUFV,@XR)	COMPARE E1 TO E2		
	10E7	F2	81	0A		7958	JE	ICT060	BRANCH IF VALUES ARE IDENTICAL		
	10EA	F2	00	0E		7959	ICT045 JC	ICT070,*-*	BRANCH IF E1 > E2 (BOTH SIGNS		
					7960	*			* POSITIVE) OR E1 < E2 (90TH		
					7961	*			* SIGNS NEGATIVE)		
					7962	*					
					7963	*	SET CONDITION STATUS CODE DEPENDING ON COMPARE OPERATION				
					7964	*					
	10ED	3C	02	117B		7965	ICT050 MVI	IZBRCN,I@CMLO	SET LOW COMPARE CONDITION		
	10F1	D0	87	29		7966	B	INTAD1(,@BR)	GO EXECUTE NEXT PSEUDO INST		
	10F4	3C	04	117B		7967	ICT060 MVI	IZBRCN,I@CMEQ	SET EQUAL COMPARE CONDITION		
	10F8	D0	87	29		7968	B	INTAD1(,@BR)	GO EXECUTE NEXT PSEUDO INST		
	10FB	3C	08	117B		7969	ICT070 MVI	IZBRCN,I@CMHI	SET HIGH COMDARE CONDITION		
	10FF	D0	87	29		7970	B	INTAD1(,@BR)	GO EXECUTE NEXT PSEUDO INST		
					7971	*					
					7972		*****				

## S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20 PAGE 130
					7974	*****		
					7975	* ENTRY ICTCSA - COMPUTE STACKED ADDRESS BASED ON INDEX		
					7976	*****		
					7977	*		
				1102	7978	ICTCSA EQU *	ICTCSA ENTRY POINT	
					7979	*		
					7980	* RUN-TIME STACK CONTAINS A SERIES OR (N+1) VIRTUAL ADDRESSES - SET UP		
					7981	* THE BINARY VALUE OR (N) CONTAINED IN THE 'CSA' INSTRUCTION OPERAND		
					7982	*		
1102	75	02	EC		7983	ICT080 L	IZXIAR(,@BR),@XR	LOAD PSEUDO INST CORE ADDRESS
1105	6C	00	F9 01		7984	MVC	IZWRK1(,@BR),I@XCNT(B@LCNN,@XR)	MOVE OPERAND (N) TO WORK
1109	7C	00	F8		7985	MVI	IZWRK1-1(,@BR),@ZERO	MAKE (N) A 2-BYTE BINARY NUMBER
					7986	*		
					7987	* TOP OR STACK CONTAINS A FLOATING POINT INDEXING VALUE - CONVERT		
					7988	* THIS TO A 2-BYTE BINARY NUMBER		
					7989	*		
110C	5F	00	EE E4		7990	ICT090 SLC	IZSTAK(,@BR),IZCL1F(@VADDR-1,@BR)	DECR STACK POINTER
1110	75	02	EE		7991	L	IZSTAK(,@BR),@XR	LOAD THE STACK POINTER
1113	C0	87	0AE3		7992	B	CAFPBS	LINK TO CONVERT INDEX TO BINARY
					7993	*		
					7994	* TEST OR CONVERSION ERROR (1 > INDEX >= 10.000) OR INDEX RANGE		
					7995	* EXCEPTION (INDEX > N) - IN EITHER CASE SET INDEX = ZERO TO REFER-		
					7996	* ENCE LEFTMOST STACKED ADDRESS IN SERIES, WHICH IN TURN REFERENCES		
					7997	* THE NEXT IN-LINE PSEUDO INSTRUCTION		
					7998	*		
1117	7D	00	5C		7999	ICT100 CLI	IZERRC(,@BR),I@NERR	IF LIST INDEX CONVERSION ERROR
111A	F2	01	07		8000	JNE	ICT110	* SKIP TO SET EXCEPTION INDEX
111D	9D	01	01 F9		8001	CLC	I@SIDX(,@XR),IZWRK1(B@LDMN,@BR)	IF INDEX IS WITHIN LIST
1121	F2	04	07		8002	JNH	ICT120	* RANGE, SO CONTINUE NORMALLY
1124	7C	00	5C		8003	ICT110 MVI	IZERRC(,@BR),I@NERR	CLEAR INTERPRETER ERROR INDR
1127	AF	01	01 01		8004	SLC	I@SIDX(,@XR),I@SIDX(B@LDMN,@XR)	ZERO INDEX FOR EXCEPTION
					8005	*		
					8006	* MOVE VIRTUAL ADDRESS (FROM SERIES) REFERENCED BY THE INDEX TO THE		
					8007	* TOP OF THE STACK - THE VIRTUAL ADDRESS SERIES AND BINARY INDEX ARE		
					8008	* DELETED FROM THE STACK		
					8009	*		
112B	5F	00	EE F9		8010	ICT120 SLC	IZSTAK(,@BR),IZWRK1(@VADDR-1,@BR)	DECR STACK POINTER PAST
112F	5F	00	EE F9		8011	SLC	IZSTAK(,@BR),IZWRK1(@VADDR-1,@BR)	* (N) STACKED VADDRS
1133	AE	01	01 01		8012	ALC	I@SIDX(,@XR),I@SIDX(B@LDMN,@XR)	DOUBLE THE INDEX VALUE
1137	2C	00	1145 01		8013	MVC	ICT130+@DD2,I@SIDX(1,@XR)	SET MOVE INST DISP EQUAL INDEX
113C	75	02	EE		8014	L	IZSTAK(,@BR),@XR	LOAD THE STACK POINTER & DECR
113F	76	02	E3		8015	A	IZCBM1(,@BR),@XR	* TO REFERENCE NEW STACK TOP
					8016	*		
1142	AC	01	00 00		8017	ICT130 MVC	I@SVAD-1(,@XR),*-(@VADDR,@XR)	MOVE INDEXED VADDR TO TOP
					8018	*		
1146	D0	87	22		8019	B	INTAD2(,@BR)	GO EXECUTE NEXT PSEUDO INST
					8020	*		
					8021	*****		
					8022	*		
					8023	* END OF LOGICAL PMC EXECUTION ROUTINES CODING		
					8024	*		

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 131
		8026		*****	
		8027	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		8028	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
		8029	*		*
		8030		*****	
		8031	*	*STATUS	*
		8032	*	VERSION 1 MODIFICATION 0	*
		8033	*		*
		8034	*	*FUNCTION	*
		8035	*	* ICBRAN CONTAINS THE RUN-TIME ROUTINES WHICH INTERPRET AND CAUSE	*
		8036	*	EXECUTION OF THE FOLLOWING PSEUDO MACHINE INSTRUCTIONS	*
		8037	*	* 'BRA' - BRANCH UNCONDITIONALLY	*
		8038	*	* 'BRC' - BRANCH ON CONDITION	*
		8039	*	* 'BRS' - BRANCH TO STACKED ADDRESS	*
		8040	*	* 'BRD' - BRANCH AND DELETE FUNCTION ENTRY	*
		8041	*	* 'BNX' - BRANCH AND SUPPRESS EXECUTION	*
		8042	*	* THE FOLLOWING DESCRIPTIONS GIVE FUNCTIONAL SPECIFICATIONS FOR	*
		8043	*	THE ROUTINES WHICH ARE USED TO EXECUTE THE PSEUDO INSTRUCTIONS	*
		8044	*	LISTED ABOVE.	*
		8045	*	* 'BRA' - BRANCH UNCONDITIONALLY (FORMAT - OP VADR)	*
		8046	*	CONTROL IS TRANSFERRED TO THAT PSEUDO INSTRUCTION WHICH	*
		8047	*	BEGINS AT VADR.	*
		8048	*	* 'BRC' - BRANCH ON CONDITION (FORMAT - OP VADR CC)	*
		8049	*	CONTROL IS TRANSFERRED TO THAT PSEUDO INSTRUCTION WHICH	*
		8050	*	BEGINS AT VADR WHEN CODE CC AGREES WITH THE CURRENT COM-	*
		8051	*	PAIRE CONDITION. OTHERWISE, CONTROL IS PASSED TO THE NEXT	*
		8052	*	SEQUENTIAL PSEUDO INSTRUCTION.	*
		8053	*	* 'BRS' - BRANCH TO STACKED ADDRESS (FORMAT - OP)	*
		8054	*	CONTROL IS TRANSFERRED TO THAT PSEUDO INSTRUCTION WHICH	*
		8055	*	BEGINS AT THE VIRTUAL ADDRESS STORED AT THE TOP OR THE	*
		8056	*	STACK. THE VIRTUAL ADDRESS IS DELETED FROM THE STACK.	*
		8057	*	* 'BRD' - BRANCH AND DELETE FUNC. ENTRY (FORMAT - OP VADR)	*
		8058	*	THE ENTRY AT THE TOP OF THE USER FUNCTION ACTIVITY TABLE	*
		8059	*	IS DELETED, AND CONTROL IS TRANSFERRED TO THAT PSEUDO	*
		8060	*	INSTRUCTION WHICH BEGINS AT VADR.	*
		8061	*	* 'BNX' - BRANCH AND SUPPRESS EXECUTION (FORMAT - OP VADR)	*
		8062	*	CONTROL IS TRANSFERRED TO THAT PSEUDO INSTRUCLION WHICH	*
		8063	*	BEGINS AT VADR, EXCEPT THE FIRST 'BRA' INSTRUCTION EN-	*
		8064	*	COUNTERED AFTER THE TRANSFER OF CONTROL IS NOT EXECUTED.	*
		8065	*	* ICBRAN ALSO CONTAINS AN ENTRY POINT (ICBSET) WHICH PERMITS	*
		8066	*	BRANCHING TO A VIRTUAL ADDRESS DEFINED DURING THE EXECUTION OF	*
		8067	*	ANOTHER PSEUDO INSTRUCTION (E.G, AN 'FCI' INSTRUCTION).	*
		8068	*		*
		8069	*	*ENTRY POINTS	*
		8070	*	* ENTRY ICBBRA - FOR EXECUTION OF THE 'BRA' INSTRUCTION.	*
		8071	*	CALLING SEQUENCE IS	*
		8072	*	B ICBBRA	*
		8073	*	* ENTRY ICBBRC - FOR EXECUTION OF THE 'BRC' INSTRUCTION.	*
		8074	*	CALLING SEQUENCE IS	*
		8075	*	B ICSBRC	*
		8076	*	* ENTRY ICBBRS - FOR EXECUTION OF THE 'BRS' INSTRUCTION.	*
		8077	*	CALLING SEQUENCE IS	*
		8078	*	B ICBBRS	*
		8079	*	ENTRY ICBBRD - FOR EXECUTION OF THE 'BRD' INSTRUCTION.	*
		8080	*	CALLING SEQUENCE IS	*
		8081	*	B ICBBRD	*

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 132
		8082	*	* ENTRY ICBSNX - FOR EXECUTION OF THE 'BNX' INSTRUCTION.	*
		8083	*	CALLING SEQUENCE IS	*
		8084	*	B ICBBNX	*
		8085	*	* ENTRY ICBSET - FOR BRANCHING TO VIRTUAL ADDRESS DEFINED	*
		8086	*	DURING EXECUTION OF ANOTHER PSEUDO INSTRUCTION.	*
		8087	*	CALLING SEQUENCE IS	*
		8088	*	B ICBSET	*
		8089	*	* EACH OF THE ABOVE ENTRY POINTS (EXCEPT FOR ICBSET) IS ACCESSED	*
		8090	*	THROUGH THE PMC EXECUTION BRANCH ADDRESS TABLE (INTBAT) IN	*
		8091	*	EXECUTIVE ROUTINE INTERP. ICBSET IS ENTERED DIRECTLY FROM THE	*
		8092	*	EXTERNAL PMC EXECUTION ROUTINE REQUIRING A BRANCHING OPERATION.	*
		8093	*	ENTRY POINTS ARE SUBJECT TO INPUT CONDITIONS DESCRIBED BELOW.	*
		8094	*		*
		8095	*	*INPUT	*
		8096	*	* IZXPAG - 1 BYTE, FOR THE EXECUTION PAGE NUMBER. THIS IS TO	*
		8097	*	CONTAIN THE VIRTUAL PAGE NUMBER FOR THE CURRENT PMC PAGE.	*
		8098	*	* IZXIAR (FOR ENTRY POINTS ICBBRA, ICBBRC, ICBBRD, ICBBNX) -	*
		8099	*	2 BYTES, FOR THE PMC ADDRESS REGISTER. THIS IS TO CONTAIN THE	*
		8100	*	CORE ADDRESS OF THE OPCODE FIELD IN THE PSEUDO INSTRUCTION	*
		8101	*	BEING EXECUTED.	*
		8102	*	* IZWRK1 (OR ENTRY POINT ICBSET) - 2 BYTES, FOR INTERPRETER	*
		8103	*	COMMON WORK AREA 1. THIS IS TO CONTAIN THE BRANCH DESTINATION	*
		8104	*	VIRTUAL ADDRESS.	*
		8105	*	* IZSTAK (FOR ENTRY POINT ICBBRS) - 2 BYTES, FOR THE RUN-TIME	*
		8106	*	STACK POINTER. THIS IS TO CONTAIN THE CORE ADDRESS OF THE	*
		8107	*	STACK LOCATION IMMEDIATELY FOLLOWING THE TOP STACKED DATA	*
		8108	*	ELEMENT, WHICH SHOULD BE A VIRTUAL ADDRESS.	*
		8109	*	* ICS090+@Q (EXTERNAL IZBRCN, ISBRCN, FOR ENTRY POINT ICBBRC) -	*
		8110	*	1 BYTE, FOR THE INTERPRETER COMPARE CONDITION CODE. THIS IS TO	*
		8111	*	CONTAIN A CODE RESULTING FROM THE LAST EXECUTION OF A COMPARE	*
		8112	*	PSEUDO INSTRUCTION (SEE ICTEST), AND IS USED IN CONJUNCTION	*
		8113	*	WITH THE 'BRC' INSTRUCTION CONDITION CODE TO FORM A BRANCHING	*
		8114	*	DECISION (SEE TABLES/WORK AREAS).	*
		8115	*	* RUN-TIME STACK (FOR ENTRY POINT ICBBRS) - THIS IS TO CONTAIN A	*
		8116	*	BRANCH DESTINATION VIRTUAL ADDRESS IN THE TOP STACK POSITION.	*
		8117	*		*
		8118	*	*OUTPUT	*
		8119	*	* IZXIAR - 2 BYTES, FOR THE PMC ADDRESS REGISTER. THIS CONTAINS	*
		8120	*	THE BRANCH DESTINATION VIRTUAL ADDRESS WHEN A VALID BRANCH HAS	*
		8121	*	BEEN EXECUTED, OR REMAINS UNCHANGED WHEN THE BRANCH IS NOT	*
		8122	*	TAKEN (A CONDITIONAL 'FALL THROUGH' IS CONSIDERED TO BE A VALID	*
		8123	*	BRANCH TO THE NEXT SEQUENTIAL INSTRUCTION).	*
		8124	*	* IZXPAG - 1 BYTE, FOR THE EXECUTION PAGE NUMBER. THIS CONTAINS	*
		8125	*	THE VIRTUAL PAGE NUMBER FOR THE BRANCH DESTINATION VIRTUAL	*
		8126	*	ADDRESS WHEN A VALID BRANCH HAS BEEN EXECUTED, OR REMAINS UN-	*
		8127	*	CHANGED WHEN THE BRANCH IS TO AN ADDRESS WITHIN THE SAME	*
		8128	*	VIRTUAL PAGE,	*
		8129	*	* IZVADR - 2 BYTES, FOR THE PAGING MODULE VIRTUAL ADDRESS PARA-	*
		8130	*	METER. THIS CONTAINS THE BRANCH DESTINATION VIRTUAL ADDRESS	*
		8131	*	WHEN A VALID BRANCH HAS BEEN EXECUTED, OR REMAINS UNCHANGED	*
		8132	*	WHEN THE BRANCH IS TO AN ADDRESS WITHIN THE SAME VIRTUAL PAGE.	*
		8133	*	* IZSTAK (AFTER ENTRY POINT ICBBRS) - 2 BYTES, FOR THE RUN-TIME	*
		8134	*	STACK POINTER. THIS CONTAINS THE CORE ADDRESS FOLLOWING THE	*
		8135	*	TOP STACK ELEMENT AFTER THE VIRTUAL ADDRESS INPUT ARGUMENT HAS	*
		8136	*	BEEN DELETED.	*
		8137	*	* IZFATP (AFTER ENTRY POINT ICBBRD) - 2 BYTES, FOR THE USER FUNC-	*



## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 06/09/20 PAGE 133

```

8138 *      TION ACTIVITY TABLE POINTER.  THIS POINTER IS DECREMENTEN BY *
8139 *      ONE TABLE ENTRY LENGTH EACH TIME 'BRD' IS EXECUTED. *
8140 *      * IZERRC - 1 BYTE, FOR THE ERROR CONDITION CODE.  THIS CONTAINS A *
8141 *      NULL CODE (I@NERR) WHEN NO ERROR CONDITION EXISTS, OR AN ERROR *
8142 *      CODE SPECIFYING THE PARTICULAR ERROR CONDITION DISCOVERED. *
8143 *      * IZARSW (AFTER ENTRY POINT ICBBNX) - 1 BYTE, FOR THE IMAGE *
8144 *      REFERENCE SWITCH.  THIS SWITCH IS SET ON (CODE @UCB) WHENEVER *
8145 *      INSTRUCTION 'BNX' IS EXECUTED. *
8146 *      * 'BRA' INSTRUCTION EXECUTION SWITCH - INTERNAL TO ICBRAN, THIS *
8147 *      SWITCH IS SET OFF ('BRA' IS DISABLED) WHENEVER INSTRUCTION *
8148 *      'BNX' IS EXECUTED, AND IS RESET ON ('BRA' IS ENABLED) AFTER *
8149 *      AN ATTEMPT IS MADE TO EXECUTE A 'BRA' INSTRUCTION USING THE *
8150 *      DISABLED ROUTINE. *
8151 * *
8152 *EXTERNAL REFERENCES *
8153 *      * INTXEC - ENTRY POINT FOR INTERPRETER PMC EXECUTION ROUTINE. *
8154 *      * INTPAG - ENTRY POINT FOR INTERPRETER NEW PAGE EXECUTION ROUTINE. *
8155 *      * IPGULK - ENTRY POINT FOR PAGING MODULE V.M. PAGE UNLOCKING RTN. *
8156 *      * INTAD3 - ENTRY POINT FOR INTERPRETER 3-BYTE PMC INCREMENT RTN. *
8157 *      * INTAD4 - ENTRY POINT FOR INTERPRETER 4-BYTE PMC INCREMENT RTN. *
8158 *      * INTERR - ENTRY POINI FOR INTERPRETER EXECUTION ERROR ROUTINE. *
8159 *      * $ENDNU - CORE ADDRESS OF FIRST BYTE FOLLOWING SYSTEM NUCLEUS. *
8160 *      * TZBASE - CORE ADDRESS FOR INTERP BASE ADDRESSABILITY. *
8161 *      * IZSTAX - 2 BYTES, FOR THE RUN-TIME STACK POINTER. *
8162 *      * IZXIAR - 2 BYTES, FOR THE PSEUDO INSTRUCTION ADDRESS REGISTER. *
8163 *      * IIXPAG - 1 BYTE, FOR THE CURRENT EXECUTION PAGE NUMBER. *
8164 *      * IZVADR - 2 BYTES, FOR PAGING MODULE VIRTUAL ADDRESS PARAMETER. *
8165 *      * IZPGN0 - 1 BYTE, FOR PAGING MODULE VIRTUAL PAGE NUMBER PARAM. *
8166 *      * IZFATP - 2 BYTES, FOR USER FUNCTION ACTIVITY TABLE POINTER. *
8167 *      * IZIRSW - 1 BYTE, FOR THE IMAGE REFERENCE SWITCH. *
8168 *      * IZERRC - 1 BYTE, FOR THE INTERPRETER EXECUTION ERROR CODE. *
8169 *      * IZWRK1 - 2 BYTES, FOR INTERPRETER COMMON WORK AREA 1. *
8170 *      * IZCLVA - 1 BYTE, FOR LENGTH OF A VIRTUAL ADDRESS. *
8171 *      * IICBN3 - 1 BYTE, FOR LENGTH OF A 3-BYTE PSEUDO INSTRUCTION. *
8172 * *
8173 *EXITS, NORMAL *
8174 *      * ENTRY POINT ICBBRA - CONTROL IS PASSED TO THE INTERPRETER *
8175 *      EXECUTIVE AT ENTRY POINT INTAD3 FOR NEXT PSEUDO INSTRUCTION *
8176 *      EYECUTION WHEN THE 'BRA' INSTRUCTION ROUTINE HAS BEEN DISABLED. *
8177 *      * ENTRY POINT ICBBRC - CONTROL IS PASSED TO THE INTERPRETER *
8178 *      EXECUTIVE AT ENTRY POINT INTAD4 FOR NEXT PSEUDO INSTRUCTION *
8179 *      EXECUTION WHEN THE 'BRC' BRANCH CONDITION CODE DOES NOT AGREE *
8180 *      WITH THE CURRENT COMPARE CONDITION CODE. *
8181 *      * INTERNAL PAGE BRANCH - CONTROL IS PASSED TO THE INTERPRETER *
8182 *      EXECUTIVE AT ENTRY POINT INTXEC WHEN THE TRANSFER OF CONTROL IS *
8183 *      TO A PSEUDO INSTRUCTION WITHIN THE CURRENT PMC EXECUTION PAGE. *
8184 *      * EXTERNAL PAGE BRANCH - CONTROL IS PASSED TO THE INTERPRETER *
8185 *      EXECUTIVE AT ENTRY POINT INTPAG WHEN THE TRANSFER OR CONTROL IS *
8186 *      TO A PSEUDO INSTRUCTION OUTSIDE THE CURRENT PMC EXECUTION PAGE. *
8187 * *
8188 *EXITS, ERROR *
8189 *      ALL ENTRY POINTS - CONTROL IS PASSED TO THE INTERPRETER EXECUTIVE *
8190 *      AT ENTRY POINT INTERR WITH PARAMETER IIERRC CONTAINING THE APPRO- *
8191 *      PRIATE ERROR MESSAGE CODE (SEE ERROR PROCEDURES). *
8192 * *
8193 *TABLES/WORK AREAS *

```



## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 134
8194	*	*		ICB090+@Q (EXTERNAL IZBRN, I&BRN) - 1 BYTE, FOR THE INTER-	*
8195	*			PRETER COMPARE CONDITION CODE. THIS FIELD IS SET WITH ONE OF	*
8196	*			THE FOLLOWING CODES WHENEVER A COMPARE INSTRUCTION ('CMF' OR	*
8197	*			'CMC') IS EXECUTED (SEE ICTEST).	*
8198	*			* CODE X'02' - LOW COMPARE	*
8199	*			* CODE X'04' - EQUAL COMPARE	*
8200	*			* CODE X'08' - HIGH COMPARE	*
8201	*				*
8202	*			*ATTRIBUTES	*
8203	*			* REUSABLE	*
8204	*			* RELOCATABLE	*
8205	*				*
8206	*			*CHARACTER CODE DEPENDENCY	*
8207	*			THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*
8208	*			INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
8209	*				*
8210	*			*NOTES	*
8211	*			ERROR PROCEDURES	*
8212	*			* ERROR 1 - THE BRANCH DESTINATION ADDRESS HAS NOT BEEN	*
8213	*			DEFINED DURING COMPILATION. AN ERROR CODE FOR THE MESSAGE	*
8214	*			'UNDEFINED LINE NUMBER REFERENCED' IS ESTABLISHED IN PARA-	*
8215	*			METER IZERRC.	*
8216	*			* ERROR 2 - NO VIRTUAL ADDRESS ARGUMENT EXISTS IN THE RUN-TIME	*
8217	*			STACK DURING EXECUTION OF A 'BRS' INSTRUCTION. AN ERROR	*
8218	*			CODE FOR THE MESSAGE 'RETURN WITHOUT ACTIVE GOSUB' IS ESTAB-	*
8219	*			LISHED IN PARAMETER IZERRC.	*
8220	*			* WHEN EITHER OR THESE CONDITIONS OCCUR, CONTROL IS PASSED TO	*
8221	*			ERROR ROUTINE INTERR TO ABORT EXECUTION.	*
8222	*				*
8223	*			REGISTER USAGE	*
8224	*			* REGISTER @BR IS EXPECTED TO CONTAIN THE INTERPRETER EXECU-	*
8225	*			TIVE ROUTINE BASE CODE ADDRESS (IZBASE) AT ICBAN ENTRY, AND	*
8226	*			RETAINS THIS ADDRESS AT EXIT.	*
8227	*			* REGISTER @XR IS NOT SAVED. IT IS USED IN ICBAN FOR GENERAL	*
8228	*			PURPOSE INDEXING.	*
8229	*				*
8230	*			SAVED/RESTORED AREAS	*
8231	*			N/A	*
8232	*				*
8233	*			MODIFICATION CONSIDERATIONS	*
8234	*			N/A	*
8235	*				*
8236	*			REQUIRED MODULES	*
8237	*			* @SYSEQ - COMMON SYSTEM EQUATES.	*
8238	*			* @FXDEQ - SYSTEM NUCLEUS ADDRESSES AND INDICATOR EQUATES.	*
8239	*			* @ERMEQ - SYSTEM ERROR MESSAGE CODE EQUATES.	*
8240	*			* \$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD. PREC. ONLY).	*
8241	*			* \$I@LEQ - INTERPRETER PARAMETER EQUATES (FOR LONG PREC. ONLY).	*
8242	*			* INTERP - INTERPRETER EXECUTIVE ROUTINE.	*
8243	*			* IPGMDL - INTERPRETER PAGING CONTROL MODULE.	*
8244	*			* IZCOMN - INTERPRETER COMMON ADDRESS REFERENCE EQUATES.	*
8245	*				*
8246	*			NOTES	*
8247	*			AN ACTUAL MACMINE INSTRUCTION OPCODE IS MODIFIED DURING EXECU-	*
8248	*			FION OF THE 'BRC' PSEUDO INSTRUCTION. A 'CLI' IS CONVERTED TO	*
8249	*			A 'TBN' OR 'TBF' TO FACILITATE CONDITION CODE TESTING AT LABEL	*

S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER	15,	MOD	00	06/09/20	PAGE	135
				8250	*		ICB090. ANY CNANGES TO ACTUAL MACHINE INSTRUCTION OPCODES FOR	*						
				8251	*		THESE ASSEMBLER INSTRUCTIONS WILL REQUIRE MODIFICATION TO THIS	*						
				8252	*		LOGIC. COMMENTS ARE PROVIDED TO INDICATE THE CONSIDERATIONS	*						
				8253	*		INVOLVED AND MECHANISM FOR CHANGING THE LOGIC.	*						
				8254	*		*****	*						

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 136
		8256		*****	
		8257	*	START OF PMC EXECUTION MODULE ICBRAN	
		8258		*****	
		8259	*		
		8260	*	START ICBRAN - ESTABLISH ADDRESSABILITY	
		8261	*		
		1149	8262	ICBRAN EQU *	START OF ICBRAN CODING
		0C60	8263	USING IZBASE,@BR	DEFINE INTERPRETER BASE ADDRESS
		8264	*		
		8265		*****	
		8266	*	ENTRY ICBBNX - DISABLE NEXT BRANCH AND BRANCH UNCONDITIONALLY	
		8267		*****	
		8268	*		
		1149	8269	ICBBNX EQU *	ICBBNX ENTRY POINT
		8270	*		
		8271	*	DISABLE THE NEXT 'BRA' INSTRUCTION FOLLOWING THIS 'BNX'	
		8272	*		
1149	3C 80 1181	8273	ICB010 MVI	ICB100+@Q,@NOP	SET 'BRA' ENABLE SWITCH OFF
		8274	*		
		8275	*	SET THE 'IMAGE REFERENCE' SWITCH ON - THIS WILL CAUSE AN ERROR	
		8276	*	CONDITION WHEN THE HEADER INSTRUCTION OF A STATEMENT OTHER THAN	
		8277	*	AN 'IMAGE' IS EXECUTED AS A RESULT OF THIS BRANCH.	
		8278	*		
114D	7C 87 7E	8279		MVI IZIRSW(,@BR),@UCB	SET 'IMAGE REFERENCE' SWITCH ON
		8280	*		
		8281	*	SKIP TO EXECUTE AN UNCONDITIONAL BRANCH OPERATION	
		8282	*		
1150	F2 87 37	8283	ICB020 J	ICB120	GO PERFORM UNCONDITIONAL BRANCH
		8284	*		
		8285		*****	

## S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 137
		8287		*****	
		8288	*	ENTRY ICEBRD - DEACTIVATE USER FUNCTION AND BRANCH UNCONDITIONALLY	
		8289		*****	
		8290	*		
		1153 8291	ICBBRD EQU *	ICBBRD ENTRY POINT	
		8292	*		
		8293	*	DELETE TOP ENTRY IN THE USER FUNCTION ACTIVITY TABLE	
		8294	*		
1153	1F 01 0DE8 E9	8295	ICB030 SLC	IZFATP, IZCLVA(@CADDR, @BR) DECR FUNC ACTIVITY TABLE PT	
		8296	*		
		8297	*	SKIP TO EXECUTE AN UNCONDITIONAL BRANCH OPERATION	
		8298	*		
1158	F2 87 2F	8299	ICB040 J	ICB120 GO PEWOEM ACONDITIONAL MANN	
		8300	*		
		8301		*****	
		8303		*****	
		8304	*	ENTRY ICBBRS - BRANCH UNCONDITIONALLY TO STACKED ADDRESS	
		8305		*****	
		8306	*		
		115B 8307	ICBBRS EQU *	ICBBRS ENTRY POINT	
		8308	*		
		8309	*	SAVE THE STACKED VIRTUAL ADDRESS (ASSUMED VALID)	
		8310	*		
115B	5F 00 EE E9	8311	ICB050 SLC	IZSTAK(, @BR), IZCLVA(@VADDR-1, @BR) DECR THE STACK POINTER	
115F	75 02 EE	8312	L	IZSTAK(, @BR), @XR LOAD THE STACK POINTER	
1162	6C 01 F9 01	8313	MVC	IZWRK1(, @BR), I@SVAD(@VADDR, @XR) SAVE TNE STACKED VADOR	
		8314	*		
		8315	*	TEST FOR INVALID VIRTUAL ADDRESS (STACK BOUNDARY CONDITION)	
		8316	*		
1166	7D 39 EE	8317	ICB060 CLI	IZSTAK(, @BR), IZSTKB-\$ENDNU IF STACKED ADDRESS IS VALID	
1169	F2 02 25	8318	JNL	ICB130 * GO PERFORM UNCOND BRANCH	
		8319	*		
		8320	*	BOUNDARY UNDERFLOW - SET 'RETURN WITHOUT ACTIVE GOSUB' ERROR	
		8321	*		
116C	7C C3 5C	8322	ICB070 MVI	IZERRC(, @BR), @@E726 SET INTERPRETER ERROR CODE	
116F	D0 87 4B	8323	B	INTERR(, @BR) GO EXECUTE THE ERROR ROUTINF	
		8324	*		
		8325		*****	

S/3 BASIC INTERPRETER INITIALIZER.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 138
		8327		*****	
		8328		* ENTRY ICBBRC - BRANCH ON COMPARE CONDITION	
		8329		*****	
		8330		*	
		1172 8331	ICBBRC EQU *	ICBBRC ENTRY POINT	
		8332		*	
		8333		* SET THE CONDITION TEST - ZONE PORTION OF THE 'BRC' INSTRUCTION	
		8334		* CONDITION CODE CONTAINS A HEX VALUE WHICH CONVERTS THE CONDITION	
		8335		* TEST (MACHINE) INSTRUCTION TO A TBN OR TBF OPERATION	
		8336		*	
1172 75 02 EC		8337	ICB080 L	IZXIAR(,@BR),@XR	LOAD 'BRC' INSTRUCTION CADDR
1175 28 02 117A 03		8338		MNZ ICB090,I@XBRC(,@XR)	SET INST ICB090 TO TBN OR TBF
		8339		*	* DEPENDING ON 'BRC' COND ZONE
		8340		*	
		8341		* TEST PSEUDO EXECUTION CONDITION - STATUS BYTE (ICB090+@Q)	
		8342		* HAS BEEN SET DURING EXECUTION OR THE PREVIOUS 'CMF' OR 'CMC' PMC	
		8343		*	
117A BD 00 03		8344	ICB090 CLI	I@XBRC(,@XR),*-*	TEST PMC BR COND VS STATUS
117D D0 90 14		8345		BF INTAD4(,@BR)	GO EXECUTE NEXT IN-LINE PMC
		8346		*	* IF BRANCH CONDITION NOT MET
		8347		*	
		8348		*****	

## S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20 PAGE 139
					8350	*****		
					8351	* ENTRY ICBBRA - BRANCH UNCONDITIONALLY		
					8352	*****		
					8353	*		
				1180	8354	ICBBRA EQU *	ICBBRA ENTRY POINT	
					8355	*		
					8356	* TEST FOR BRANCH EXECUTION DISABLED (BY PREVIOLS 'BNY' INSTRUCTION)		
					8357	*		
1180	F2	00	07		8358	ICB100 JC ICB120,*-*	IF 'BRA' INSTRUCTION ENABLED	
1181					8359	ORG ICB100+@Q	* GO PERFORM BRANCH EXECUTION	
1181	87			1181	8360	DC AL1(@UCB)	INITIALIZE JUMP INSTRUCTION	
1183					8361	ORG ICB100+@INST3	* FOR ENABLED CONDITION	
					8362	*		
					8363	* 'BRA' DISABLED - RE-ENABLE, THEN BRANCH TO EXECUTE NEXT IN-LINE PMC		
					8364	*		
1183	3C	87	1181		8365	ICB110 MVI ICB100+@Q,@UCB	ENABLE 'BRA' PMC EXECUTION	
1187	D0	87	1B		8366	B INTAD3(,@BR)	GO EXECUTE NEXT PSEUDO INST	
					8367	*		
					8368	* ACCESS THE PSEUDO INSTRUCTION VIRTUAL ADDRESS OPERAND		
					8369	*		
118A	75	02	EC		8370	ICB120 L IZXIAR(,@BR),@XR	LOAD BRANCH PMC CORE ADDRESS	
118D	6C	01	F9 02		8371	MVC IZWRK1(,@BR),I@XVAD(@VADDR,@XR)	MOVE INST OPERAND TO WORK	
					8372	*		
					8373	* TEST FOR AN UNRESOLVED (UNDEFINED) BRANCH ADDRESS		
					8374	*		
1191	7D	56	F8		8375	ICB130 CLI IZWRK1-1(,@BR),@VENTA	IF PMC VADDR OPERAND DEFINED	
1194	F2	02	06		8376	JNL ICB150	* GO CONTINUE BRANCH EXECUTION	
					8377	*		
					8378	* UNRESOLVFD ADDRESS - SET 'UNDEFINED LINE NUMBER REFERENCED' ERROR		
					8379	*		
1197	7C	B4	5C		8380	ICB140 MVI IZERRC(,@BR),@@E700	SET INTERPRETER ERROR CODE	
119A	D0	87	4B		8381	B INTERR(,@BR)	GO TERMINATE ON LINE NO. ERROR	
					8382	*		
					8383	*****		

## S/3 BASIC INTERPRETER INITIALIZER.

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 140
				8385	*****	
				8386	* ENTRY ICBSET - ESTABLISH ADDRESS FOR NEXT PMC TO BE EXECUTED	
				8387	*****	
				8388	*	
			119D	8389	ICBSET EQU * ICBSET ENTRY POINT	
				8390	*	
				8391	* ADDRESS DEFINED - TEST FOR A BRANCH INTERNAL TO CURRENT EXEC PAGE	
				8392	*	
119D	5D	00 F8 01		8393	ICB150 CLC IZWRK1-1(,@BR),IZXPAG(1,@BR) IF NOT BRANCH TO CURR PAGE	
11A1	F2	01 07		8394	JNE ICB170 * GO SET BRANCH TO NEW PAGE	
				8395	*	
				8396	* INTERNAL BRANCH - ESTABLISH THE NEW EXECUTION CORE ADDRESS	
				8397	*	
11A4	5C	00 EC F9		8398	ICB160 MVC IZXIAR(,@BR),IZWRK1(@CADDR-1,@BR) SET EXEC PAGE NEW DISP	
11A8	D0	87 2D		8399	B INTXEC(,@BR) GO START EXECUTION AT NEW CADDR	
				8400	*	
				8401	* EXTERNAL BRANCH - UNLOCK CURRENT EXEC PAGE FROM CORE VIRTAL MEMORY	
				8402	*	
11AB	1C	00 1449 01		8403	ICB170 MVC IZPGNO,IZXPAG(1,@BR) RESTORE CURRENT PMC PAGE NO.	
11B0	C0	87 1350		8404	B IPGULK LINK TO RELEASE CURR PMC PAGE	
				8405	*	
				8406	* ESTABLISH THE NEW EXECUTION VIRTUAL ADDRESS	
				8407	*	
11B4	5C	00 01 F8		8408	ICB180 MVC IZXPAG(,@BR),IZWRK1-1(1,@BR) SET NEW EXECUTION PAGE NO.	
11B8	1C	01 144A F9		8409	MVC IZVADR,IZWRK1(@VADDR,@BR) SET PAGING PARM FOR NEW PAGE	
11BD	D0	87 08		8410	B INTPAG(,@BR) BRANCH TO GET NEW PMC PAGE	
				8411	*	
				8412	*****	
				8413	*	
				8414	* END OF BRANCH EXECUTION PMC ROUTINES CODING	
				8415	*	



ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 141
		8417		*****	
		8418	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		8419	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
		8420	*		*
		8421		*****	
		8422	*	STATUS	*
		8423	*	VERSION 1 MODIFICATION 0	*
		8424	*		*
		8425	*	FUNCTION	*
		8426	*	* ICLOOP CONTAINS THE RUN-TIME ROUTINES WHICH INTERPRET AND CAUSE	*
		8427	*	EXECUTION OF THE FOLLOWING PSEUDO MACHINE INSTRUCTIONS -	*
		8428	*	* 'FOR' - INITIATE 'FOR' LOOP	*
		8429	*	* 'NXT' - PERFORM 'NXT' STEP	*
		8430	*	* THE FOLLOWING DESCRIPTIONS GIVE FUNCTIONAL SPECIFICATIONS FOR	*
		8431	*	THE ROUTINES WHICH ARE USED TO EXECUTE THE PSEUDO INSTRUCTIONS	*
		8432	*	LISTED ABOVE.	*
		8433	*	* 'FOR' - INITIATE 'FOR' LOOP (FORMAT - OP VADR)	*
		8434	*	THIS INSTRUCTION IS ALWAYS PAIRED WITH A TRAILING 'NXT'	*
		8435	*	INSTRUCTION. VADR IS THE VIRTUAL ADDRESS OF THE LOOP CON-	*
		8436	*	TROL VARIABLE. THE FLOATING POINT VALUE THIRD IN THE	*
		8437	*	STACK (THE LOOP CONTROL INITIAL VALUE) IS SAVED IN A CON-	*
		8438	*	TROL VARIABLE WORK AREA. THE FLOATING POINT VALUES SECOND	*
		8439	*	IN THE STACK AND AT THE TOP OF THE STACK THE FINAL VALUE	*
		8440	*	AND INCREMENT, RESPECTIVELY) ARE STORED IN A 'DWA' DEFINED	*
		8441	*	WORK AREA FOLLOWING THE 'NXT' INSTRUCTION IN THE PMC	*
		8442	*	SEQUENCE. THE THREE FLOATING POINT VALUES ARE DELETED	*
		8443	*	FROM THE STACK AND CONTROL IS TRANSFERRED TO THE 'NXT'	*
		8444	*	INSTRUCTION ROUTINE SUCH THAT CONTROL VARIABLE RETRIEVAL	*
		8445	*	AND INCREMENTATION ARE BYPASSED.	*
		8446	*	* 'NXT' - PERFORM 'NEXT' STEP (FORMAT - OP VADR)	*
		8447	*	THIS INSTRUCTION IS ALWAYS PAIRED WITH A PRECEDING 'FOR'	*
		8448	*	INSTRUCTION AND ALWAYS PRECEDES A 'DWA' DEFINED WORK AREA	*
		8449	*	CONTAINING THE FINAL VALUE AND INCREMENT FOR THE LOOP.	*
		8450	*	* THE LOOP CONTROL VARIABLE STORED AT 'FOR' INSTRUCTION	*
		8451	*	OPERAND AN IS PLACED IN A CONTROL VARIABLE WORK AREA AND	*
		8452	*	MODIFIED USING THE LOOP INCREMENT.	*
		8453	*	* WHEN THE WORKING VALUE OF THE CONTROL VARIABLE EXCEEDS THE	*
		8454	*	FINAL VALUE, CONTROL IS TRANSFERRED TO THAT PSEUDO INSTRU-	*
		8455	*	TION WHICH BEGINS AT 'NXT' OPERAND VADR. OTHERWISE, THE	*
		8456	*	WORKING VALUE OF THE CONTROL VARIABLE IS STORED AT 'FOR'	*
		8457	*	OPERAND VADR, AND CONTROL IS PASSED TO THE FIRST INSTRU-	*
		8458	*	TION WHICH FOLLOWS THE LIMIT/INCREMENT WORK AREA.	*
		8459	*		*
		8460	*	ENTRY POINTS	*
		8461	*	* ENTRY ICLFOR - FOR EXECUTION OF THE 'FOR' INSTRUCTION.	*
		8462	*	CALLING SEQUENCE IS	*
		8463	*	B ICLFOR	*
		8464	*	* ENTRY ICLNXT - FOR EXECUTION OF THE 'NXT' INSTRUCTION.	*
		8465	*	CALLING SEQUENCE IS	*
		8466	*	B ICLNXT	*
		8467	*	* EACH OR THE ABOVE ENTRY POINTS IS ACCESSED THROUGH THE PMC	*
		8468	*	EXECUTION BRANCH ADDRESS TABLE (INTBAT) IN EXECUTIVE ROUTINE	*
		8469	*	INTERP, AND IS SUBJECT TO INPUT CONDITIONS DESCRIBED BELOW.	*
		8470	*		*
		8471	*	INPUT	*
		8472	*	* IZSTAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER. THIS IS TO	*

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 142
		8473	*	CONTAIN THE CORE ADDRESS OF THE FIRST AVAILABLE STACK LOCATION.	*
		8474	*	* IZXIAR - 2 BYTES, FOR THE PMC ADDRESS REGISTER. THIS IS TO	*
		8475	*	CONTAIN THE CORE ADDRESS OF THE OPCODE FIELD IN THE PSEUDO	*
		8476	*	INSTRUCTION BEING EXUCUTED.	*
		8477	*	* 'FOR' INSTRUCTION OPERAND - 2 BYTES, CONTAINS THE VIRTUAL	*
		8478	*	ADDRESS OF THE 'FOR' LOOP CONTROL VARIABLE.	*
		8479	*	* 'NXT' INSTRUCTION OPERAND - 2 BYTES, CONTAINS THE VIRTUAL	*
		8480	*	ADDRESS OF THE PSEUDO INSTRUCTION FOLLOWING THE CURRENT	*
		8481	*	'FOR/'NEXT' LOOP STATEMENTS.	*
		8482	*	* RUN-TIME STACK (FOR ENTRY POINT ICLFOR) - THIS IS TO CONTAIN	*
		8483	*	THREE UNPACKED FLOATING POINT VALUES -	*
		8484	*	* THIRD STACK VALUE - LOOP INITIAL VALUE	*
		8485	*	* SECOND STACK VALUE - LOOP FINAL VALUE	*
		8486	*	* TOP STACK VALUE - LOOP INCREMENT	*
		8487	*	* VIRTUAL MEMORY (FOR ENTRY POINT ICLNVT) - THIS CONTAINS THE	*
		8488	*	CONTROL VARIABLE VALUE BEFORE INCREMENTATION.	*
		8489	*	* 'FOR'/'NXT' WORK AREA (FOR ENTRY POINT ICLNXT) - FOR TWO UN-	*
		8490	*	PACKED FLOATING POINT VALUES. THIS 'DWA' DEFINED AREA CONTAINS*	*
		8491	*	THE LOOP FINAL VALUE AND INCREMENT WHEN THE LOOP HAS BEEN IN	*
		8492	*	TIALIZED DURING 'FOR' INSTRUCTION EXECUTION. THE FIRST BYTE IN*	*
		8493	*	THIS AREA IS SET TO BINARY ZERO WHEN THE LOOP HAS NOT BEEN SO	*
		8494	*	INITIALIZED, CAUSING ERROR 2 (SEE ERROR PROCEDURES).	*
		8495	*		*
		8496	*	*OUTPUT	*
		8497	*	T IZ5TAK (AFTER ENTRY POINT ICLFOR) - 2 BYTES, FOR THE RUN-TIME	*
		8498	*	STACK POINTER. THIS CONTAINS THE CORE ADDRESS OF THE FIRST	*
		8499	*	AVAILABLE STACK LOCATION AFTER THE THREE FLOATING POINT VALUES	*
		8500	*	HAVE BEEN DELETED.	*
		8501	*	* IZXIAR - 2 BYTES, FOR THE PMC ADDRESS REGISTER. WHEN THE CON-	*
		8502	*	TROL VARIABLE (DURING 'FOR' INSTRUCTION EXECUTION) OR THE INCRE-	*
		8503	*	MENTED CONTROL VARIABLE (DURING 'NXT' INSTRUCTION EXECUTION)	*
		8504	*	DOES NOT EXCEED THE FINAL LOOP VALUE, THIS CONTAINS THE CORE	*
		8505	*	ADDRESS OF THE FIRST PSEUDO INSTRUCTION FOLLOWING THE	*
		8506	*	'FOR'/'NXT' WORK AREA. WHEN THE FINAL VALUE HAS BEEN EXCEEDED,*	*
		8507	*	IXXIAR CONTAINS THE CORE ADDRESS OF THE 'NXT' OPCODE FIELD.	*
		8508	*	* IZERRC - 1 BYTE, FOR THE ERROR CONDITION CODE. THIS CONTAINS A*	*
		8509	*	NULL CODE (I@NERR) WHEN NO ERROR CONDITION EXISTS, OR AN ERROR	*
		8510	*	CODE SPECIFYING THE PARTICULAR ERROR CONDITION DISCOVERED.	*
		8511	*	* IZRESW - 1 BYTE, FOR THE STATEMENT RECURSION ERROR SWITCH.	*
		8512	*	WHEN THE FINAL LOOP VALUE IS NOT EXCEEDED, THIS IS SET TO CODE	*
		8513	*	@NOP TO DISABLE RECURSION ERROR PROCESSING DURING THE NEXT	*
		8514	*	'STH' INSTRUCTION EXECUTION.	*
		8515	*	* 'FOR'/'NXT' WORK AREA - FOR TWO UNPACKED FLOATING POINT VALUES.*	*
		8516	*	WHEN THE FINAL LOOP VALUE IS NOT EXCEEDED, THIS 'DWA' DEFINED	*
		8517	*	AREA CONTAINS THE LOOP FINAL VALUE AND INCREMENT. WHEN THE	*
		8518	*	FINAL VALUE IS EXCEEDED, THE FIRST BYTE IN THIS AREA IS SET TO	*
		8519	*	BINARY ZERO BEFORE LEAVING THE LOOP.	*
		8520	*	* LOCK AND READ ONLY INDICATOR TAKE - WHENEVER THE 'FOR'/'NXT'	*
		8521	*	WORK AREA IS AFFECTED, THE APPROPRIATE INDICATOR IN THIS PAGING*	*
		8522	*	MODULE TABLE IS SET ON (USING ENTRY POINT IPGMOD) TO DEFINE THE*	*
		8523	*	CURRENT PMC CORE PAGE AS HAVING BEEN MODIFIED.	*
		8524	*	* VIRTUAL MEMORY - THE CONTROL VARIABLE IN VIRTUAL MEMORY IS	*
		8525	*	MODIFIED TO THE VALUE USED DURING LOOP EXECUTION. WHEN EXEC-	*
		8526	*	TION CONTROL IS PASSED TO ICBRA AT LOOP TERMINATION, THE CON-	*
		8527	*	TROL VARIABLE RETAINS THE VALUE USED DURING FINAL LOOP EXECU-	*
		8528	*	TION. WHEN A CONTROL VARIABLE EXCEPTION OCCURS BEFORE A	*

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 143
		8529	*	SINGLE LOOP EXECUTION, THE CONTROL VARIABLE REMAINS UNCHANGED	*
		8530	*	IN VIRTUAL MEMORY.	*
		8531	*		*
		8532	*	EXTERNAL REFERENCES	*
		8533	*	* FDIADD - ENTRY POINT FOR FLOATING POINT ADD ROUTINE.	*
		8534	*	* FDISUB - ENTRY POINT FOR FLOATING POINT SUBTRACT ROUTINE.	*
		8535	*	* ISTACK - ENTRY POINT FOR INTERPRETER ELEMENT STACKING RTN.	*
		8536	*	* IUSTAK - ENTRY POINT FOR INTERPRETER ELEMENT UNSTACKING RTN.	*
		8537	*	* CPUFLT - ENTRY POINT FOR FLOATING POINT VALUE UNPACKING ROUTINE.	*
		8538	*	* CUPFLT - ENTRY POINT FOR FLOATING POINT VALUE PACKING ROUTINE.	*
		8539	*	* IPGMOD - ENTRY POINT FOR PAGING MODULE V.M. PAGE MODIFY ROUTINE.	*
		8540	*	* INTXEC - ENTRY POINT FOR INTERPRETER PMC EXECUTION ROUTINE.	*
		8541	*	* ICBBRA - ENTRY POINT FOR 'BRA' INSTRUCTION EXECUTION ROUTINE.	*
		8542	*	* INTERR - ENTRY POINT FOR INTERPRETER EXECUTION ERROR ROUTINE.	*
		8543	*	* IZBASE - CORE ADDRESS FOR INTERP BASE ADDRESSABILITY.	*
		8544	*	* IZSTKB - CORE ADDRESS OF RUN-TIME STACK LEFTMOST BYTE.	*
		8545	*	* IZSTAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER.	*
		8546	*	* IZXIAR - 2 BYTES, FOR THE PSEUDO INSTRUCTION ADDRESS REGISTER.	*
		8547	*	* IZXPAG - 1 BYTE, FOR THE CURRENT EXECUTION PAGE NUMBER.	*
		8548	*	* IZVADR - 2 BYTES, FOR PAGING MODULE VIRTUAL ADDRESS PARAMETER.	*
		8549	*	* IZPGNO - 1 BYTE, FOR PAGING MODULE VIRTUAL PAGE NUMBER PARAM.	*
		8550	*	* IZRESW - 1 BYTE, FOR THE RECURSIVE STATEMENT ERROR SWITCH.	*
		8551	*	* IZERRC - 1 BYTE, FOR THE INTERPRETER EXECUTION ERROR CODE.	*
		8552	*	* IZCBN3 - 1 BYTE, FOR LENGTH OF A 3-BYTE PSEUDO INSTRUCTION.	*
		8553	*		*
		8554	*	EXITS, NORMAL	*
		8555	*	* NORMAL EXECUTION - CONTROL IS PASSED TO THE INTERPRETER AT	*
		8556	*	ENTRY POINT INTVEC WHEN LOOP EXECUTION IS TO BE CONTINUED	*
		8557	*	(I.E, WHEN THE CONTROL VARIABLE DOES NOT EXCEED THE LIMIT),	*
		8558	*	* TERMINAL EXECUTION - CONTROL IS PASSED TO BRANCH PMC ENTRY	*
		8559	*	POINT ICBBRA WHEN LOOP EXECUTION IS TO BE TERMINATED (I.E. WHEN	*
		8560	*	THE CONTROL VARIABLE EXCEEDS THE LIMIT).	*
		8561	*		*
		8562	*	EXITS, ERROR	*
		8563	*	ALL ENTRY POINTS - CONTROL IS PASSED TO THE INTERPRETER EXECUTIVE	*
		8564	*	AT ENTRY POINT INTERR WITH PARAMETER IZERRC CONTAINING THE APPRO-	*
		8565	*	PRIATE ERROR MESSAGE CODE (SEE ERROR PROCEDURES).	*
		8566	*		*
		8567	*	TABLES/WORK AREAS	*
		8568	*	AN IN-LINE PMC SAVE AREA (DEFINED WITH A 'DWA' INSTRUCTION) IS	*
		8569	*	UTILIZED TO STORE THE INITIALLY SPECIFIED LOOP CONTROL VARIABLE	*
		8570	*	LIMIT AND INCREMENT VALUES. THIS AREA IS LARGE ENOUGH TO CONTAIN	*
		8571	*	TWO PACKED FLOATING POINT VALUES, AND IMMEDIATELY FOLLOWS THE	*
		8572	*	LOOP 'NXT' PSEUDO INSTRUCTION IN VIRTUAL MEMORY.	*
		8573	*		*
		8574	*	ATTRIBUTES	*
		8575	*	* REUSABLE	*
		8576	*	* RELOCATABLE	*
		8577	*		*
		8578	*	CHARACTER CODE DEPENDENCY	*
		8579	*	THE OPERATION OF THIS MODULE DEPENDS UPON THE FOLLOWING PROPER-	*
		8580	*	TIES OF INTERNAL PRESENTATION OF THE EXTERNAL CHARACTER SET.	*
		8581	*	* NUMERIC CHARACTERS 0 THROUGH 9 ARE PRESUMED TO BE CODED SUCH	*
		8582	*	THAT THE HIGH ORDER FOUR BITS CONTAIN A SIGN ZONE SPECIFI-	*
		8583	*	CATION.	*
		8584	*	THE SPECIFIC INSTRUCTIONS (INSTRUCTION SEQUENCE) WHICH REQUIRE	*

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 144
		8585	*	MODIFICATION IF THESE PROPERTIES OF THE CHARACTER SET ARE CHANGED	*
		8586	*	MAY BE IDENTIFIED BY -	*
		8587	*	* THE 6 INSTRUCTIONS BEGINNING AT LABEL ICL100	*
		8588	*	COMMENTS ARE PROVIDED TO INDICATE THE CONSIDERATIONS INVOLVED AND	*
		8589	*	MECHANISMS FOR CHANGING THE CODE.	*
		8590	*		*
		8591	*	*NOTES	*
		8592	*	ERROR PROCEDURES	*
		8593	*	* ERROR 1 - A SUBTRACTION ERROR OCCURS (SEE FDISUB) DURING THE	*
		8594	*	LIMIT EXCEPTION CALCULATION. THE ERROR CODE ESTABLISHED BY	*
		8595	*	FDISUB IS RETAINED IN PARAMETER IZERRC.	*
		8596	*	* ERROR 2 - THE LOOP CONTROL WORK AREA WAS NOT BEEN INITIAL-	*
		8597	*	IZED WITH LIMIT AND STEP VALLES PRIOR TO EXECUTION OF THE	*
		8598	*	'NXT' PSEUDO INSTRUCTION. AN ERROR CODE EOR THE MESSAGE	*
		8599	*	'INVALID FOR LOOP EXECUTION' IS ESTABLISHED IN PARAMETER	*
		8600	*	IZERRC.	*
		8601	*	* ERROR 3 - AN ADDITION ERROR OCCURS (SEE FDIADD) DURING THE	*
		8602	*	CONTROL VARIABLE INCREMENTATION CALCULATION. THE ERROR CODE	*
		8603	*	ESTABLISHED BY FDIADD IS RETAINED IN PARAMETER IZERRC.	*
		8604	*	* WHEN ANY OF THESE CONDITIONS EXIST, CONTROL IS PASSED TO	*
		8605	*	ERROR ROUTINE INTERR TO ABORT EXECUTION.	*
		8606	*		*
		8607	*	REGISTER USAGE	*
		8608	*	* REGISTER @BR IS EXPECTED TO CONTAIN THE INTERPRETER EXEC-	*
		8609	*	TIVE ROUTINE BASE CORE ADDRESS (IZBASE) AT ICLOOP ENTRY, AND	*
		8610	*	RETAINS THIS ADDRESS AT EXIT.	*
		8611	*	* REGISTER @XR IS NOT SAVED. IT IS USED IN ICLOOP FOR GENERAL	*
		8612	*	PURPOSE INDEXING.	*
		8613	*		*
		8614	*	SAVED/RESTORED AREAS	*
		8615	*	N/A	*
		8616	*		*
		8617	*	MODIFICATION CONSIDERATIONS	*
		8618	*	N/A	*
		8619	*		*
		8620	*	REQUIRED MODULES	*
		8621	*	* @SYSEQ - COMMON SYSTEM EQUATES.	*
		8622	*	* @ERMEQ - SYSTEM ERROR MESSAGE CODE EQUATES.	*
		8623	*	* \$B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.	*
		8624	*	* \$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD. PREC. ONLY).	*
		8625	*	* \$I@LEQ - INTERPRETER PARAMETER EQUATES (FOR LONG PREC. ONLY).	*
		8626	*	* FDIADD - FLOATING POINT ADD/SUBTRACT ROUTINE.	*
		8627	*	* ISTACK - INTERPRETER ELEMENT STACKING ROUTINE.	*
		8628	*	* IUSTAK - INTERPRETER ELEMENT UNSTACKING ROUTINE.	*
		8629	*	* CPUFLT - FLOATING POINT VALUE UNPACKING ROUTINE.	*
		8630	*	* CUPFLT - FLOATING POINT VALUE PACKING ROUTINE.	*
		8631	*	* INTERP - INTERPRETER EXECUTIVE ROUTINE.	*
		8632	*	* ICBRAN - INTERPRETER BRANCH PMC EXECUTION ROUTINE.	*
		8633	*	* IPSMDL - INTERPRETER PAGING CONTROL MODULE.	*
		8634	*	* IZCOMN - INTERPRETER COMMON ADDRESS REFERENCE EQUATES.	*
		8635	*		*
		8636	*	OTHER	*
		8637	*	N/A	*
		8638	*	*****	*

## ICOOP - S/3 BASIC INTERPRETER FOR/NXT PMC EXEC RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 145
		8640		*****	
		8641	*	START OF PMC EXECUTION MODULE ICLOOP	
		8642		*****	
		8643	*		
		8644	*	START ICLOOP - ESTABLISH ADDRESSABILITY	
		8645	*		
		11C0	8646	ICLOOP EQU *	START OF ICLOOP CODING
		0C60	8647	USING IZBASE,@BR	DEFINE INTERPRETER BASE ADDRESS
		8648	*		
		8649		*****	
		8650	*	ENTRY ICLFOR - INITIALIZE AND BEGIN FOR/NEXT LOOP EXECUTION	
		8651		*****	
		8652	*		
		11C0	8653	ICLFOR EQU *	
		8654	*		
		8655	*	INITIALIZE LOOP ENTRY EXECUTION - STACK CONTAINS 3 FLOATING POINT	
		8656	*	VALUES: - INITIAL VALUE	
		8657	*	- FINAL VALUE	
		8658	*	- CONTROL VARIABLE INCREMENT	
		8659	*		
11C0	4F 00 EE 1279	8660	ICL010	SLC IZSTAK(,@BR),ICLL3F(@VADDR-1)	DECREMENT STACK POINTER
11C5	75 02 EE	8661		L IZSTAK(,@BR),@XR	LOAD THE STACK POINTER
11C8	75 01 EC	8662		L IZXIAR(,@BR),@BR	LOAD 'FOR' PSEUDO INST CADDR
		8663	*		
		8664	*	SAVE CONTROL VARIABLE LIMIT & STEP IN IN-LINE PMC WORK AREA	
		8665	*		
11CB	6C 0F 17 17	8666	ICL020	MVC ICLFV2(,@BR),I@RSE3(2*I@LUFV,@XR)	SAVE LIMIT/STEP IN PMC
11CF	0C 00 1449 0C61	8667		MVC IZPGNO,IZXPAG(1)	RESTORE CURRENT PMC PAGE NO.
11D5	C0 87 1349	8668		B IPGMOD	LINK TO SET PAGE MODIFY SWITCH
11D9	F2 87 3A	8669		J ICL080	SKIP TO BYPASS 'NEXT' ROUTINE
		8670	*		
		8671		*****	



ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 146
		8673		*****	
		8674	*	ENTRY ICLNXT - INCREMENT CONTROL VARIABLE & TEST LOOP CONTINUATION	
		8675		*****	
		11DC 8676	ICLNXT EQU *	ICLNXT ENTRY POINT	
		8677	*		
		8678	*	ACCESS 'FOR' PSEUDO INSTRUCTION PRECEDING 'NXT' INSTRUCTION	
		8679	*		
11DC 5F 00 EC E0		8680	ICL030 SLC	IZXIAR(,@BR),IZCBN3(@CADDR-1,@BR) DECR POINTER TO 'FOR'	
11E0 75 01 EC		8681	L	IZXIAR(,@BR),@BR LOAD 'FOR' PSEUDO INST CADDR	
		8682	*		
		8683	*	TEST FOR INVALID LOOP ENTRY - THIS CONDITION EXISTS WHEN 'NXT' IS	
		8684	*	ENCOUNTERED WITHOUT THE ASSOCIATED 'FOR' HAVING BEEN EXECUTED	
		8685	*		
11E3 7D 00 08		8686	ICL040 CLI	ICLFSW(,@BR),@ZERO IF CURRENT LOOP IS ACTIVE	
11E6 F2 01 08		8687	JNE	ICL050 * GO CONTINUE LOOP EXECUTION	
11E9 3C C0 0CBC		8688	MVI	IZERRC,@E723 SET 'INVAL LOOP EXECUTION' ERR	
11ED C0 87 0CAB		8689	B	INTERR GO TERMINATE ON LOOP ERROR	
		8690	*		
		8691	*	STACK THE CONTROL VARIABLE CURRENT VALUE	
		8692	*		
11F1 35 02 0D4E		8693	ICL050 L	IZSTAK,@XR LOAD THE STACK POINTER	
11F5 1C 01 144A 02		8694	MVC	IZVADR,I@XVAD(@VADDR,@BR) SET PAGING PARM FOR CTL VAR	
11FA C0 87 0B50		8695	B	ISTACK LINK TO STACK TOIL CONTROL VAR	
11FE C0 87 0A27		8696	B	CPUFLT LINK TO UNPACK THE CONTROL VAR	
		8697	*		
		8698	*	INCREMENT THE CONTROL VARIABLE BY THE LOOP STEP VALUE	
		8699	*		
1202 9C 07 0F 17		8700	ICL060 MVC	I@RSE2(,@XR),ICLFV2(I@LUFV,@BR) STACK THE LOOP STEP VALUE	
1206 C0 87 075D		8701	B	FDIADD LINK TO ADD STEP TO CONTROL VAR	
120A 3D 00 0CBC		8702	CLI	IZERRC,I@NERR IF ADDITION ERROR CONDITION	
120E C0 01 0CAB		8703	BNE	INTERR * GO TERMINATE ON ARITH ERROR	
		8704	*		
		8705	*	ESTABLISH LOOP LIMIT AND STEP IN STACK FOR LIMIT TEST	
		8706	*		
1212 9C 0F 17 17		8707	ICL070 MVC	I@RSE3(,@XR),ICLFV2(2*I@LUFV,@BR) STACK LIMIT/STEP VALUES	
		8708	*		
		8709		*****	

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 147
		8711		*****	
		8712	*	LOOP CONTROL VARIABLE LIMIT EXCEPTION TEST ROUTINE	
		8713		*****	
		8714	*		
		8715	*	SAVE INITIAL OR INCREMENTED VALUE OF THE CONTROL VARIABLE	
		8716	*		
1216	C2 01 0C60	8717	ICL080 LA	IZBASE,@BR	RESTORE INTERPRETER BASE REG
121A	2C 07 0638 07	8718		MVC IZSTKB-1,I@RSE1(I@LUFV,@XR)	SAVE STACKED CTRL VAR VALUE
		8719	*		
		8720	*	SUBTRACT LOOP FINAL VALUE (LIMIT) FROM CONTROL VARIABLE VALUE	
		8721	*		
121F	C0 87 0751	8722	ICL090 B	FDISUB	LINK TO SUB LIMIT FROM CTL VAR
1223	7D 00 5C	8723		CLI IZERRC(,@BR),I@NERR	IF SUBTRACTION ERROR CONDITION
1226	D0 01 4B	8724		BNE INTERR(,@BR)	* GO TERMINATE ON ARITH ERROR
		8725	*		
		8726	*	TEST FOR LOOP CONTINUATION OR LIMIT EXCEPTION	
		8727	*		
1229	BD F0 01	8728	ICL100 CLI	I@MANL(,@XR),B@DEC0	IS CONTROL VARIABLE EQUAL LIMIT
122C	F2 81 27	8729		JE ICL140	* SKIP TO CONTINUE LOOP EXEC
122F	28 00 123A 17	8730		MZZ ICL110+@Q,I@RSE3(,@XR)	MOVE STEP SIGN ZONE TO CMP INST
1234	28 03 123A 07	8731		MNN ICL110+@Q,I@SIGN(,@XR)	EQUALIZE NUMERIC ZONES FOR CMP
1239	BD 00 07	8732	ICL110 CLI	I@SIGN(,@XR),*-*	IT NO CTL VAR LIMIT EXCEPTION
123C	F2 01 17	8733		JNE ICL140	* SKIP TO CONTINUE LOOP EXEC
		8734	*		
		8735		*****	



ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20 PAGE 148
				8737		*****		
				8738	*	CONTROL VARIABLE LIMIT EXCEPTION ROUTINE		
				8739		*****		
				8740	*			
				8741	*	DEACTIVATE EXECUTION FOR CURRENT LOOP		
				8742	*			
123F	75	02	EC	8743	ICL120	L	IZXIAR(,@BR),@XR	LOAD 'FOR' PSEUDO INST CADDR
1242	BC	00	08	8744		MVI	ICLFSW(,@XR),@ZERO	SET LOOP ACTIVATION SWITCH OFF
1245	1C	00	1449 01	8745		MVC	IZPGNO,IZXPAG(1,@BR)	RESTORE CURRENT PMC PAGE NO.
124A	C0	87	1349	8746		B	IPGMOD	LINK TO SET PAGE MODIFY SWITCH
				8747	*			
				8748	*	EXECUTE PSEUDO BRANCH TO END OF CURRENT FOR/NEXT LOOP		
				8749	*			
124E	5E	00	EC E0	8750	ICL130	ALC	IZXIAR(,@BR),IZCBN3(@VADDR-1,@BR)	INCR PMC POINT TO 'NXT'
1252	C0	87	1180	8751		B	ICBBRA	GO EXECUTE BRANCH TO EXIT LOOP
				8753		*****		
				8754	*	CONTROL VARIABLE STORAGE AND LOOP EXECUTION ROUTINE		
				8755		*****		
				8756	*			
				8757	*	UNSTACK NEW CONTROL VARIABLE VALUE TO VIRTUAL MEMORY		
				8758	*			
1256	75	02	EC	8759	ICL140	L	IZXIAR(,@BR),@XR	LOAD 'FOR' PSEUDO PST CADDAI
1259	2C	01	144A 02	8760		MVC	IZVADR,I@XVAD(@VADDR,@XR)	SET PAGING PARM FOR CTL VAR
125E	75	02	EE	8761		L	IZSTAK(,@BR),@XR	LOAD THE STACK POINTER
1261	8C	07	07 0638	8762		MVC	I@RSE1(,@XR),IZSTKB-1(I@LUFV)	STACK NEW CTRL VAR VALUE
1266	C0	87	0A85	8763		B	CUPFLT	LINK TO PACK TFE CONTROL VAR
126A	C0	87	0BB0	8764		B	IUSTAK	LINK TO UNSTICK THE CONTROL VAR
				8765	*			
				8766	*	BEGIN LOOP EXECUTION FOR CURRENT VALUE OR CONTROL VARIABLE		
				8767	*			
126E	4E	00	EC 127A	8768	ICL150	ALC	IZXIAR(,@BR),ICLFSZ(@CADDR-1)	INCR PMC PT TO 1ST LOOP OPC
1273	7C	80	89	8769		MVI	IZRESW(,@BR),@NOP	DISABLE RECURSION ERROR EXEC
1276	D0	87	2D	8770		B	INTXEC(,@BR)	GO EXECUTE 1ST LOOP PSEUDO INST.
				8771	*			
				8772		*****		

ICOOP - S/3 BASIC INTERPRETER FOR/NXT PMC EXEC RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 149
		8774		*****	
		8775		* FOR/NXT PMC ROUTINE CONSTANTS	
		8776		*****	
		8777		*	
1279 18		1279	8778	ICLL3F DC AL(@VADDR-1)(3*I@LUFV) LENGTH OF 3 FLT POINT VALUES	
127A 18		127A	8779	ICLFSZ DC AL(@CADDR-1)(B@LFOR+B@LNXT+B@LDWA+2*I@LUFV) 'FOR' SEQ LNG	
		8781		*****	
		8782		* FOR/NXT PMC ROUTINE EQUATES	
		8783		*****	
		8784		*	
	0008	8785		ICLFSW EQU B@LFOR+B@LNXT+B@LDWA DISP FOR LOOP ACTIVATFON SWITCH	
	0017	8786		ICLFV2 EQU B@LFOR+B@LNXT+B@LDWA+2*I@LUFV-1 DISP FOR LIMIT/STEP AREA	
		8787		*****	
		8788		*	
		8789		* END OF FOR/NXT PMC EXECUTION ROUTINES CODING	
		8790		*	

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20	PAGE 150
		8792		*****			
		8793	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
		8794	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
		8795	*				*
		8796		*****			*
		8797	*	*STATUS			*
		8798	*	VERSION 1 MODIFICATION 0			*
		8799	*				*
		8800	*	*FUNCTION			*
		8801	*	* ICMEX CONTAINS INTERFACES WHICH ACCESS AND PASS CONTROL TO			*
		8802	*	VIRTUAL MEMORY RESIDENT ROUTINES WHICH INTERPRET AND EXECUTE			*
		8803	*	THE FOLLOWING PSEUDO MACHINE INSTRUCTIONS -			*
		8804	*	* 'FCI' - FUNCTION CALL, INDIRECT			*
		8805	*	* 'SD0' - STACK DOPE VECTOR (NO REDIMENSIONING)			*
		8806	*	* 'SD1' - STACK DOPE VECTOR (REDIMENSION 1)			*
		8807	*	* 'SD2' - STACK DOPE VECTOR (REDIMENSION 2)			*
		8808	*	* 'GET' - INPUT DATA ELEMENT			*
		8809	*	* 'PUT' - OUTPUT DATA ELEMENT			*
		8810	*	* 'INI' - INITIATE KEYBOARD INPUT			*
		8811	*	* 'ADF' - ACTIVATE EXTERNAL DATA FILE			*
		8812	*	* 'RSR' - RESTORE INTERNAL DATA FILE POINTER			*
		8813	*	* 'RST' - RESET INTERNAL DATA FILE POINTER			*
		8814	*	* 'CLS' - CLOSE EXTERNAL DATA FILE			*
		8815	*	* 'PRS' - PRINT AND SPACE CARRIER			*
		8816	*	* 'PRU' - PRINT USING IMAGE			*
		8817	*	* IN GENERAL, CONTROL IS PASSED TO THE APPROPRIATE V.M. ROUTINE,			*
		8818	*	AND THE V.M. ROUTINE RETURNS CONTROL TO THE INTERFACE. ERRORS			*
		8819	*	WHICH OCCUR DURING PMC EXECUTION ARE HANDLED BY THE INTERFACE			*
		8820	*	AFTER CONTROL HAS BEEN RETURNED.			*
		8821	*				*
		8822	*	*ENTRY POINTS			*
		8823	*	* ENTRY ICMFCI - FOR INTERFACING TO THE 'FCI' INSTRUCTION EXECU-			*
		8824	*	TION ROUTINE. CALLING SEQUENCE IS			*
		8825	*	B ICMFCI			*
		8826	*	* ENTRY ICMVDN - FOR INTERFACING TO THE ARRAY DOPE VECTOR STACK-			*
		8827	*	ING INSTRUCTIONS EXECUTION MODULE. CALLING SEQUENCE IS			*
		8828	*	B ICMVDN			*
		8829	*	* ENTRY ICMVFO - FOR INTERFACING TO THE INPUT/OUTPUT INSTRUCTIONS			*
		8830	*	EXECUTION MODULE. CALLING SEQUENCE IS			*
		8831	*	B ICMVFO			*
		8832	*	* EACH OF THE ABOVE ENTRY POINTS IS ACCESSED THROUGH THE PMC			*
		8833	*	EXECUTION BRANCH ADDRESS TABLE (INTBAT) IN EXECUTIVE ROUTINE			*
		8834	*	INTERP, AND IS SUBJECT TO INPUT CONDITIONS DESCRIBED BELOW.			*
		8835	*				*
		8836	*	*INPUT			*
		8837	*	* REFER TO EXECUTION PMC MODULES IDIFNC, IDDVST AND IDFILE.			*
		8838	*				*
		8839	*	*OUTPUT			*
		8840	*	* REFER TO EXECUTION PMC MODULES IDIFNC, IDDVST AND IDFILE.			*
		8841	*	* IZERRC - 1 BYTE, FOR THE ERROR CONDITION CODE. THIS CONTAINS			*
		8842	*	A NULL CODE (I@NERR) WHEN NO ERROR CONDITION EXISTS, OR AN			*
		8843	*	ERROR CODE SPECIFYING THE PARTICULAR ERROR CONDITION DISCOVERED.			*
		8844	*				*
		8845	*	*EXTERNAL REFERENCES			*
		8846	*	* V\$IFCI - VIRTUAL ENTRY ADDRESS FOR IDIFNC, 'FCI' EXECUTION RTN.			*
		8847	*	* V\$ISDN - VIRTUAL ENTRY ADDRESS FOR IDDVST, DOPE VEC. PMC MODULE.			*

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 151
		8848	*	* V\$IFIO - VIRTUAL ENTRY ADDRESS FOR IDFILE, I/O PMC EXEC. MODULE.	*
		8849	*	* IPGCAL - ENTRY POINT FOR PAGING MODULE V.M. PROGRAM CALL RTN.	*
		8850	*	* ICBSET - ENTRY POINT FOR ICBRAN EXECUTION CONTROL BRANCH RTN.	*
		8851	*	* INTERR - ENTRY POINT FOR INTERPRETER 3-BYTE PMC INCREMENT RTN.	*
		8852	*	* INTERR - ENTRY POINT FOR INTERPRETER EXECUTION ERROR ROUTINE.	*
		8853	*	* IZBASE - CORE ADDRESS FOR INTERPRETER BASE ADDRESSABILITY.	*
		8854	*	* IZDMSW - 1 BYTE, FOR THE DATA MATCHING SWITCH. WHEN THIS	*
		8855	*	SWITCH IS SET ON (IZDMSW = @NOP), UNSTACKING ROUTINE IUSTAK IS	*
		8856	*	SET TO MATCH DATA TYPES WHEN STORING DATA IN VIRTUAL MEMORY.	*
		8857	*	* IZERRC - 1 BYTE, FOR THE INTERPRETER EXECUTION ERROR CODE.	*
		8858	*		*
		8859	*	*EXITS, NORMAL	*
		8860	*	* ENTRY POINT ICVECI - CONTROL IS PASSED TO THE BRANCH EXEOTION	*
		8861	*	ROUTINE AT ENTRY POINT ICBSET FOR CONTINUED USER FUNCTION	*
		8862	*	EXECUTION WHEN NO ERROR CONDITION HAS OCCURRED.	*
		8863	*	* ENTRY PONT ICVSDN - CONTROL IS PASSED TO THE INTERPRETER AT	*
		8864	*	ENTRY POINT INTAD3 FOR NEXT PSEUDO INSTRUCTION EXECUTION WHEN	*
		8865	*	NO ERROR CONDITION HAS OCCURRED.	*
		8866	*	* ENTRY POINT ICVFIO - CONTROL IS PASSED TO THE INTERPRETER AT	*
		8867	*	ENTRY POINT INTDA1, INTAD2 OR INTAD3 (DEPENDING ON THE CURRENT	*
		8868	*	PMC LENGTH) FOR NEXT PSEUDO INSTRUCTION EXECUTION WHEN NO ERROR	*
		8869	*	CONDITION HAS OCCURRED.	*
		8870	*		*
		8871	*	*EXITS, ERROR	*
		8872	*	ALL ENTRY POINTS - CONTROL IS PASSED TO THE INTERPRETER EXECUTIVE	*
		8873	*	AT ENTRY POINT INTERR, WITH PARAMETER IZERRC CONTAINING THE	*
		8874	*	APPROPRIATE ERROR CODE, WHEN AN ERROR CONDITION HAS OCCURRED	*
		8875	*	DURING V.M-RESIDENT ROUTINE EXECUTION (SEE ERROR PROCEDURES).	*
		8876	*		*
		8877	*	*TABLES/WORK AREAS	*
		8878	*	N/A	*
		8879	*		*
		8880	*	*ATTRIBUTES	*
		8881	*	* REUSABLE	*
		8882	*	* RELOCATABLE	*
		8883	*		*
		8884	*	*CHARACTER CODE DEPENDENCY	*
		8885	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*
		8886	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
		8887	*		*
		8888	*	*NOTES	*
		8889	*	ERROR PROCEDURES	*
		8890	*	ALL PMC EXECUTION FUNCTIONS EXECUTED THROUGH ICVMEX ARE PER-	*
		8891	*	FORMED USING VIRTUAL MEMORY RESIDENT SUBROUTINES. WHEN AN	*
		8892	*	ERROR OCCURS DURING EXECUTION OF ONE OF THESE ROUTINES, AN	*
		8893	*	APPROPRIATE ERROR CODE IS LEFT IN INTERPRETER PARAMETER IZERRC	*
		8894	*	THIS PARAMETER IS TESTED WHEN CONTROL IS RETURNED TO ICVMEX	*
		8895	*	AND, IF AN ERROR HAS BEEN DISCOVERED, CONTROL IS PASSED TO THE	*
		8896	*	INTERPRETER EXECUTIVE AT ENTRY POINT INTERR.	*
		8897	*		*
		8898	*	REGISTER USAGE	*
		8899	*	* REGISTER @BR IS EXPECTED TO CONTAIN THE INTERPRETER EXECU-	*
		8900	*	TIVE ROUTINE BASE CORE ADDRESS (IZBASE) AT ICVMEX ENTRY, AND	*
		8901	*	RETAINS THIS ADDRESS AT EXIT.	*
		8902	*	* REGISTER @XR IS NOT SAVED. IT IS USED IN ICVMEX FOR GENERAL	*
		8903	*	PURPOSE INDEXING.	*

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	06/09/20	PAGE 152
		8904	*				*
		8905	*	SAVE/RESTORED AREAS			*
		8906	*	N/A			*
		8907	*				*
		8908	*	MODIFICATION CONSIDERATIONS			*
		8909	*	N/A			*
		8910	*				*
		8911	*	REQUIRED MODULES			*
		8912	*	* @SYSEQ - COMMON SYSTEM EQUATES.			*
		8913	*	* \$V\$EQU - VIRTUAL MEMORY FIXED ADDRESS EQUATES.			*
		8914	*	* \$I@SEQ - INTERPRETER PARARETER EQUATES (FOR STD. PREC. ONLY).			*
		8915	*	* \$I@LEQ - INTERPRETER PARAMETER EQUATES (FOR LONG PREC. ONLY).			*
		8916	*	* IUSTAK - INTERPRETER ELEMENT UNSTACKING ROUTINE.			*
		8917	*	* INTERP - INTERPRETER EXECUTIVE ROUTINE.			*
		8918	*	* ICBRAN - INTERPRETER BRANCH PMC EXECUTION ROUTINE.			*
		8919	*	* IPGMDL - INTERPRETER PAGING CONTROL MODULE.			*
		8920	*	* IZCOMN - INTERPRETER COMMON ADDRESS REFERENCE EQUATES.			*
		8921	*				*
		8922	*	OTHER			*
		8923	*	N/A			*
		8924	*	*****			*

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 153
		8926		*****	
		8927	*	START OF PMC EXECUTION INTERFACE MODULE ICMVEX	
		8928		*****	
		8929	*		
		8930	*	START ICMVEX - ESTABLISH ADDRESSABILITY	
		8931	*		
		127B 8932	ICMVEX EQU *	START OF ICMVEX CODING	
		0C60 8933	USING IZBASE,@BR	DEFINE INTERPRETER BASE ADDRESS	
		8935		*****	
		8936	*	ENTRY ICMVFCI - PERFORM INDIRECT FUNCTION CALL	
		8937		*****	
		8938	*		
		127B 8939	ICMVFCI EQU *	ICMVFCI ENTRY POINT	
		8940	*		
		8941	*	ESTABLISH LINKAGE TO USER FUNCTION DEFINED BY INSTRUCTION OPERAND	
		8942	*		
127B C0 87 130B		8943	ICV010 B	IPGCAL	LINK TO ESTABLISH FUNC LINKAGE
127F 1B00		1280 8944	DC	AL(@VADDR)(V\$IFCI)	FUNC LINKAGE RTN VADDR ENTRY PT
		8945	*		
		8946	*	CONTROL RETURNS WITH THE FUNCTION ARGUMENT STORED FOR EXECUTION.	
		8947	*	RETURN LINKAGE ESTABLISHED, AND BRANCHING PARAMETERS SET FOR THE	
		8948	*	1ST PSEUDO INSTRUCTION OF THE REFERENCED USER FUNCTION	
		8949	*		
1281 7D 00 5C		8950	ICV020 CLI	IZERRC(,@BR),I@NERR	IF NO FUNC CALL ERROR CONDITION
1284 C0 81 119D		8951	BE	ICBSET	* GO EXECUTE 1ST FUNCTION INST
		8952	*		
		8953	*	FUNCTION CALL ERROR CONDITION - TERMINATE EXECUTION	
		8954	*		
1288 D0 87 4B		8955	ICV030 B	INTERR(,@BR)	GO TERMINATE ON FUNC CALL ERROR
		8956	*		
		8957		*****	

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 154
		8959		*****	
		8960		* ENTRY ICVSDN - STACK/MODIFY AN ARITHMETIC ARRAY DOPE VECTOR	
		8961		*****	
		8962		*	
		128B 8963	ICVSDN EQU *	ICVSDN ENTRY POINT	
		8964		*	
		8965		* EXECUTE DOPE VECTOR STACKING OPERATION - STACK CONTAINS 1 FLOATING	
		8966		* POINT VALUE (FOR VECTOR ALLAY REOMENSIONING), 2 FLOATING POINT	
		8967		* VALUES (FOR MATRIR ARRAY REDIMENSIONING), OR NO VALUE (WHEN ARRAY	
		8968		* REDIMENSIONIN5 HAS NOT PEEN SPECIFIED)	
		8969		*	
128B C0 87 130B		8970	ICV040 B	IPGCAL	LINK TO STACKIMODIFY DOPE VECTR
128F 1900		1290 8971	DC	AL(@VADDR)(V\$ISDN)	D/V STACKING RTN VADDR ENTRY PT
		8972		*	
		8973		* CONTROL RETURNS WITH (MODIFIED) DOPE VECTOR AT TOP OF THE STACK -	
		8974		* BRANCH TO EXECUTE NEXT PSEUDO INSTRUCTION UNLESS REDIRENSIONING	
		8975		* ERROR HAS OCCOMED	
		8976		*	
1291 7D 00 5C		8977	ICV050 CLI	IZERRC(,@BR),I@NERR	IF NO REDIMENSIONING ERROR
1294 D0 81 1B		8978	BE	INTAD3(,@BR)	* SO EXECUTE NEXT PSEUDO INST
		8979		*	
		8980		* REDIMENSIONING ERROR CONDIION - BRANCH TO TERMINATE EXECUTION	
		8981		*	
1297 D0 87 4B		8982	ICV060 B	INTERR(,@BR)	GO TERMINATE ON REDIM ERROR
		8983		*	
		8984		*****	



ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 155
			8986	*****	
			8987	* ENTRY ICVFIO - PERFORM FILE INPUT/OUTPUT OPERATION	
			8988	*****	
			8989	*	
		129A	8990	ICVFIO EQU *	ICVFIO ENTRY POINT
			8991	*	
			8992	* EXECUTE I/O OPERATION DEFINED BY INSTRUCTION OPCODE	
			8993	*	
129A	3C 80 0BC1		8994	ICV070 MVI IZDMSW,@NOP	ENABLE DATA TYPE MATCHING
129E	C0 87 130B		8995	B IPGCAL	LINK TO PERFORM I/O OPERATION
12A2	1A00	12A3	8996	DC AL(@VADDR)(V\$IFIO)	FILE I/O RTNS VADDR ENTRY POINT
12A4	3C 87 0BC1		8997	MVI IZDMSW,@UCB	DISABLE DATA TYPE MATCHING
			8998	*	
			8999	* CONTROL RETURNS WITHN @XR SET FOR 1, 2 OR 3-BYTE PSEUDO INSTRUCTION	
			9000	* INCREMENT ROUTINE ADDRESS - BRANCH TO EXECUTE NEXT PSEUDO INSTRUCTION	
			9001	* UNLESS AN I/O ERROR HAS OCCURRED	
			9002	*	
12A8	7D 00 5C		9003	ICV080 CLI IZERRC(,@BR),I@NERR	IF NO I/O ERROR CONDITION
12AB	E0 81 00		9004	BE ICVADN(,@XR)	* GO EXECUTE NEXT PSEUDO INST
			9005	*	
			9006	* I/O EROR CONDITION - BRANCH TO TERMINATE EXECUTION	
			9007	*	
12AE	D0 87 4B		9008	ICV090 B INTERR(,@BR)	GO TERMINATE ON I/O ERROR
			9009	*	
			9010	*****	
			9012	*****	
			9013	* VIRTUAL MEMORY PMC ROUTINES INTERFACE EQUATES	
			9014	*****	
			9015	*	
		0000	9016	ICVADN EQU 0	DISP FOR EXEC CONTINUATION BR
			9017	*	
			9018	*****	
			9019	*	
			9020	* END OF VIRTUAL MEMORY PMC ROUTINES INTERFACE	

## IPGMDL - PAGING MODULE INTERFACES

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 156
			9022	*	*****	*
			9023	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
			9024	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
			9025	*		*
			9026	*	*****	*
			9027	*	*STATUS	*
			9028	*	VERSION 1 MODIFICATION 0	*
			9029	*		*
			9030	*	*FUNCTION	*
			9031	*	THIS THE IPGMDL (PAGING MODULE) INTERFACES BETWEEN CORE ROUTINES	*
			9032	*	(INCLUDING VIRTUAL MEMORY PAGES PRESENTLY IN CORE) AND	*
			9033	*	VIRTUAL MEMORY. IT PROVIDES THE CAPABILITY OF ADDRESSING VIRTUAL	*
			9034	*	MEMORY DIRECTLY AND PROVIDES SUBROUTINE COMMUNICATION WITHIN VM.	*
			9035	*	SEVERAL OPTIONS GIVE USER CONTROL OVER THE REPLACEMENT PROCESS.	*
			9036	*	THE PAGING MOD HAS VARIOUS ENTRY POINTS. THESE DESCRIPTIONS	*
			9037	*	APPEAR BELOW UNDER 'ENTRY POINTS'.	*
			9038	*		*
			9039	*	IPGCVA PICKS UP THE VIRTUAL ADDRESS AT LOCATION IPGVAD AND CHECKS	*
			9040	*	TO SEE IF THE PAGE IS IN CORE. IF THE PAGE IS IN CORE THE USAGE	*
			9041	*	VALUE FOR THE PAGE IS SET TO THE VALUE OF THE REFERENCE COUNTER	*
			9042	*	AND STORED IN LOCATION IPGCAD AND RETURN IS MADE.	*
			9043	*	IF THE PAGE REFERENCED IS NOT IN CORE, A BRANCH IS MADE TO IPGSLT.	*
			9044	*	IPGSLT SELECTS A PAGE TO REPLACE AND BRANCHES TO IPGRTRN.	*
			9045	*	PGRTRN WRITES THE SELECTED PAGE TO DISK IF IT HAS BEEN MODIFIED	*
			9046	*	IN CORE AND BRANCHES TO IPGRED.	*
			9047	*		*
			9048	*	*ENTRY POINTS:	*
			9049	*	IPGKAL - UNLOCK LINE PRINTER BUFFER.	*
			9050	*	IPGRTRN - SUBROUTINE RETURN ROUTINE FOR VM, UNLOCKS RETURNING	*
			9051	*	PAGE UNLESS THE RETURN IS TO ANOTHER POINT IN THE	*
			9052	*	SAME PAGE, RESTORES @BR OF THE CALLER AND RETURNS TO	*
			9053	*	CALLER.	*
			9054	*	IPGCAL - LINK AND LOCK CALLED PAGE	*
			9055	*	IPGLBR - CALL IPGCVA & POINT @BR TO CORE PAGE	*
			9056	*	IPGLXR - CALL IPGCVA & POINT @XR TO CORE PAGE	*
			9057	*	IPGMOD - SET PAGE MODIFIED (READ-ONLY) BIT ON.	*
			9058	*	IPGULK - RESET PAGE LOCKED BIT.	*
			9059	*	IPGLOK - SET PAGE LOCKED BIT ON.	*
			9060	*	IPGRED - READS THE REFERENCED PAGE INTO THE CORE SPACE	*
			9061	*	ALLOCATED AND RETURNS TO IPGCVA	*
			9062	*	IPGMOV - READ/WRITE VIRTUAL PAGE CALCULATE DISK ADDRESS	*
			9063	*	IPGSLT - SELECTS THE CORE PAGE WITH THE LOWEST USAGE VALUE TO	*
			9064	*	REPLACED,	*
			9065	*		*
			9066	*	*NOTES	*
			9067	*	THIS IS THE IPGMDL AS MENTIONED IN THE BASIC PLM PUB NR LY31-0001	*
			9068	*		*
			9069	*	*****	*
			9070	*		*
			9071	*	* PAGING MODULE EQUATES	*
			9072	*		*
0002			9073	IPGLUV EQU	2	LENGTH OF USAGE VALUE
000A			9074	IPGNMX EQU	I@NCPG	MAXIMUM NUMBER OF PAGES
0001			9075	IPGLBT EQU	X'01'	PAGE LOCK BIT IN IPGLRT ENTRY
0002			9076	IPGRBT EQU	X'02'	PAGE READ-ONLY BIT IN IPGLRT
			9077	*		*

## IPGMDL - PAGING MODULE INTERFACES

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 06/09/20 PAGE 157

```

          9078 * IPGKAL - UNLOCK LINE PRINTER BUFFER - ENTRY TO IPSCAL          1-4
          9079 *** NOTE: IPGKAL IS NOT USED BY DECALL                          1-4
          9080 *
12B1 34 08 1345      12B1 9081 IPGKAL EQU * ENTRY POINT                      1-4
          9082 ST IPG200+@OP2,@ARR SAVE CALLING RETURN ADDRESS          1-4
12B5 F2 00 12        9083 IPG170 JC IPG175,*-* JUMP LINE PRINTER BUFFER UNLK1-4
12B6                9084 ORG IPG170+@Q SET BYPASS UNLOCK BUFFER          1-4
12B6 87              12B6 9085 DC AL1(@UCB) UNCONDITIONAL JUMP SET          1-4
12B8                9086 ORG IPG170+@INST3 ORG FOR NEXT INSTRUCTION          1-4
12B8 C1 E2 12CA      9087 TIO IPG175,@PBUSY TEST FOR PRINTER BUSY          1-4
12BC 0C 01 144A 12D2 9088 MVC IPGVAD(2),IPGBFR GET LINE PRINTER BUFFER VADDR1-4
12C2 C0 87 1350      9089 B IPGULK UNLOCK LINE PRINTER BUFFER          1-4
12C6 3C 87 12B6      9090 MVI IPG170+@Q,@UCB SET LINE PRINTER BUFFER          1-4
          9091 * * UNLOCKED INDICATOR          1-4
12CA 35 08 1345      9092 IPG175 L IPG200+@OP2,@ARR RESTORE ARR          1-4
12CE F2 87 3A        9093 J IPGCAL JUMP TO CALL ROUTINE          1-4
12D1 4F00            12D2 9094 IPGBFR DC AL2(X'4F00') LINE PRINTER BUFFER VADDR 1-4
          9095 *
          9096 * PGRTRN - UNLOCK RETURNING PAGE AND RETURN TO CALLER
          9097 *
12D3 0F 01 130E 1308 12D3 9098 IPGRTN EQU * ENTRY POINT
          9099 SLC IPGCAL+@OP1,IPGB04(@CADDR) UNSTACK
12D9 35 01 130E      9100 L IPGCAL+@OP1,@BR POINT TO STACK TOP
12DD 1C 01 1306 02   9101 MVC IPG160+@OP1,@CADDR(@CADDR,@BR) FETCH RETURN ADDR
12E2 74 08 02        9102 ST @CADDR(,@BR),@ARR GET RETURNING PAGE POINTER
12E5 4F 01 02 1308   9103 SLC @CADDR(@CADDR,@BR),IPGB04 ADJUST POINTER TO PAGE
12EA 3C 20 12FE      9104 IPG100 MVI IPG120+@D1,@PGCSZ GET END OF CORE IN PAGES
12EE 1F 00 12FE 01   9105 SLC IPG120+@D1,1(1,@BR) GET PAGE NO
12F3 1C 01 1302 00   9106 MVC IPG140+@OP1,0(2,@BR) GET CALLER BASE REG
12F8 C2 01 15E1      9107 LA IPGLRT-1,@BR POINT TO IPGLRT
12FC 7B 01 00        9108 IPG120 SBF *-*(@BR),IPGLBT UNLOCK PAGE
12FF C2 01 0000      9109 IPG140 LA *-*,@BR RESTORE CALLER PAGE BASE
1303 C0 87 0000      9110 IPG160 B *-* RETURN TO CALLER PAGE
1307 0004            1308 9111 IPGB04 DC AL(@CADDR)(@CADDR+@CADDR) STACK SIZE
1309 0002            130A 9112 IPGB02 DC AL(@REGL)(@CADDR) PARAMETER LENGTH
          9113 *
          9114 * IPGCAL - LINK AND LOCK CALLED PAGE
          9115 *
130B 34 01 0000      130B 9116 IPGCAL EQU * ENTRY POINT
          9117 ST *-*,@BR PUT @BR IN STACK
130D                9118 ORG IPGCAL+@D1 INITIALIZE STORE ADDR
130D 15CB            130E 9119 DC AL(@CADDR)(IPGSTK+1) FIRST STACK LOCATION
130F 35 01 130E      9120 L IPGCAL+@OP1,@BR POINT TO STACK LOCATION
1313 74 08 02        9121 ST @CADDR(,@BR),@ARR PUT ARR IN STACK
1316 4E 01 02 130A   9122 ALC @CADDR(@CADDR,@BR),IPGB02 MAKE STACK VALUE RETURN ADDR.
131B 0E 01 130E 1308 9123 ALC IPGCAL+@OP1(2),IPGB04 UPDATE STACK POINTER
1321 3C 01 13B4      9124 MVI IPG360+@Q,IPGLBT MAKE PAGE LOCKED
1325 3C 80 13D3      9125 MVI IPG460+@Q,@NOP SET UP BRANCH TO CALLED PAGE
          9126 * FALL THRU TO PGLDBR

```

## IPGMDL - PAGING MODULE INTERFACES

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 06/09/20 PAGE 158

```

          9128 *
          9129 * PGLDBR - CALL IPGCVA & POINT @BR TO CORE PAGE
          9130 *
1329 3C 80 137F      1329 9131 IPGLBR EQU    *          ENTRY POINT
132D F2 87 04      9132          MVI    IPG260+@Q,@NOP      ENABLE LDBR CODE
          9133          J      IPG180
          9134 *
          9135 * PGLDXR - CALL IPGCVA & POINT @XR TO CORE PAGE
          9136 *
1330 3C 80 138D      1330 9137 IPGLXR EQU    *          ENTRY POINT
          9138          MVI    IPG280+@Q,@NOP      ENABLE LDBR CODE
          9139 *
          9140 * SET PARAMETER FOLLOWING CALL FOR PGLDBR/PGLDXR/PGCALL
          9141 *
1334 36 08 0464      9142 IPG180 A      $C0001,@ARR      POINT TO PARM
1338 34 08 1345      9143          ST      IPG200+@OP2,@ARR  SET MVC ADDR
133C 36 08 0464      9144          A      $C0001,@ARR      POINT TO RETURN POINT
1340 0C 01 144A 0000  9145 IPG200 MVC    IPGVAD(2),*-*      GET VADDR FROM CALLING SEQUENCE
1346 F2 87 0F        9146          J      IPGCVA          GO TO CVAD
          9147 *
          9148 * PGMDFY - MAKE SET BIT THE READ ONLY BIT
          9149 *
1349 3C 02 13B4      1349 9150 IPGMOD EQU    *          ENTRY POINT
134D F2 87 08      9151          MVI    IPG360+@Q,IPGRBT    MAKE BIT SET THE READ ONLY BIT
          9152          J      IPGCVA          GO TO CVAD
          9153 *
          9154 * PGUNLK - MAKE INSTRUCTION A SBF --- GENERATE OPCODE
          9155 *
1350 3C 00 0000      1350 9156 IPGULK EQU    *          ENTRY POINT
1351          9157          MVI    *-* ,*-*      SET
1351 BB 00 00      9158          ORG    IPGULK+@Q          * SBF
1352          9159          SBF    *-* (,@XR),*-*      * OPCODE
1352 13B3      9160          ORG    IPGULK+@D1          * IN
          1353 9161          DC      AL(@CADDR)(IPG360)    * IPG360
          9162 * FALL THRU TO PGLOCK
          9163 *
          9164 * PGLOCK - MAKE BIT SET THE LOCK BIT
          9165 *
1354 3C 01 13B4      1354 9166 IPGLOK EQU    *          ENTRY POINT
          9167          MVI    IPG360+@Q,IPGLBT    MAKE BIT SET THE LOCK BIT
          9168 * FALL THRU TO PGCVA

```

## IPGMDL - PAGING MODULE INTERFACES

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 06/09/20 PAGE 159

```

9170 *
9171 * PGCVAD - CONVERT VIRTUAL TO CORE ADDRESS
9172 *
13A8 9173 USING IPGBAS,@BR
1358 9174 IPGCVA EQU * ENTRY POINT
1358 34 01 13D1 9175 ST IPG440+@OP1,@BR SAVE BR
135C C2 01 13A8 9176 LA IPGBAS,@BR SET BASE REGISTER
1360 74 08 2D 9177 ST IPG460+@OP1(,@BR),@ARR SAVE RETURN ADDRESS
1363 74 02 25 9178 ST IPG420+@OP1(,@BR),@XR SAVE INDEX REG
9179 *
9180 * FIND PAGE TABLE ENTRY AND TEST FOR PAGE IN CORE.
9181 *
1366 C2 02 14CA 9182 IPG220 LA IPGTBL,@XR POINT TO PAGE TABLE
136A 76 02 A1 9183 A IPGVPG(,@BR),@XR POINT TO PAGE TABLE ENTRY
136D BD 00 00 9184 CLI 0(,@XR),@ZERO ZERO MEANS PAGE IS NOT IN CORE
1370 D0 81 AD 9185 BE IPGSLT(,@BR) GO TO IPGSLT IF NECESSARY
9186 *
9187 * CALCULATE CORE ADDRESS - IPGCAD = (PGSYSZ-IPGCPG)*256+PAGE DISP.
9188 * IPGRED RETURNS TO IPG220
9189 *
1373 7C 20 A3 9190 IPG240 MVI IPGCPG(,@BR),@PGCSZ PUT CORE SIZE IN CORE PAGE NO.
1376 6F 00 A3 00 9191 SLC IPGCPG(1,@BR),0(,@XR) SUBTRACT PAGE NUMBER
137A 5C 00 A4 A2 9192 MVC IPGCAD(,@BR),IPGVAD(1,@BR) GET PAGE DISP FOR CADDR
9193 *
9194 * SET LP PGLDBR/PGLDXR - SKIP THIS CODE IF NOT APPLICABLE
9195 *
137E F2 87 0B 9196 IPG260 JC IPG280,@UCB INITIALLY SKIP
1381 7C 00 29 9197 MVI IPG440+@OP1(,@BR),@ZERO ZERO DISP OF @BR SAVE
1384 5C 00 28 A3 9198 MVC IPG440+@D1(,@BR),IPGCPG(1,@BR) SET CPGNO IN BR SAVE LOC
1388 3C 87 137F 9199 MVI IPG260+@Q,@UCB RESTORE JC Q CODE
138C F2 87 08 9200 IPG280 JC IPG320,@UCB INITIALLY SKIP
138F 5C 01 25 A4 9201 IPG300 MVC IPG420+@OP1(,@BR),IPGCAD(@CADDR,@BR) SET CADDR IN @XR SA
1393 3C 87 138D 9202 MVI IPG280+@Q,@UCB RESTORE JC Q CODE
9203 *
9204 * STORE REFERENCE COUNTER IN APPROPRIATE USAGE TABLE LOCATION
9205 *
1397 6C 00 01 00 9206 IPG320 MVC IPG340+@OP1(,@BR),0(1,@XR) PUT USAGE TABLE INDEX/2 IN MVC
139B 7C 00 00 9207 MVI IPG340+@D1(,@BR),@ZERO ZERO HIGH ORDER BITS
139E 5E 01 01 01 9208 ALC IPG340+@OP1(,@BR),IPG340+@OP1(@CADDR,@BR) GET INDEX
13A2 5E 01 01 96 9209 ALC IPG340+@OP1(,@BR),IPGUTA(@CADDR,@BR) ADD TABLE ADDRESS
13A6 1C 01 0000 A6 9210 IPG340 MVC *-*,IPGUVL(IPGLUV,@BR) MOVE USAGE COUNT
9211 *
9212 * SET APPROPRIATE IPGLRT BIT
9213 *
13AB 6C 00 0D 00 9214 MVC IPG360+@D1(1,@BR),0(,@XR) GET IPGLRT DISP
13AF C2 02 15E1 9215 LA IPGLRT-1,@XR POINT TO IPGLRT
13B3 BA 00 00 9216 IPG360 SBN *-*(,@XR),*-* WHOLE INST MODIFIABLE
13B6 7C 00 00 9217 IPG380 MVI *-*(,@BR),*-* RESTORE
13B7 9218 ORG IPG380+@Q * SBN
13B7 BA 00 00 9219 SBN *-*(,@XR),*-* * OP CODE
13B8 9220 ORG IPG380+@D1 * IN
13B8 0B 13B8 9221 DC AL1(IPG360-IPGBAS) * IPG360
13B9 7C 00 0C 9222 MVI IPG360+@Q(,@BR),@ZERO RESTORE IPG360 Q CODE
9223 *
9224 * UPDATE REFERENCE COUNTER AND TEST FOR OVERFLOW
9225 *

```

IPGMDL - PAGING MODULE INTERFACES

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	06/09/20	PAGE 160
	13BC	4E 01 A6	0464		9226	ALC	IPGUVL(IPGLUV,@BR),\$C0001	UPDATE COUNTER		
	13C1	F2 20 06			9227	JNOL	IPG420	JUMP NO OVERFLOW		
	13C4	0F 13 15FF	15FF		9228	IPG400 SLC	IPGUVT+IPGNMX*IPGLUV-1(IPGNMX*IPGLUV),IPGUVT+IPGNMX*2-1			
					9229	*				
					9230	* RESTORE REGISTERS AND RETURN TO CALLER				
					9231	*				
	13CA	C2 02	0000		9232	IPG420 LA	*-*,@XR	RESTORE XR		
	13CE	C2 01	0000		9233	IPG440 LA	*-*,@BR	RESTORE BR		
	13D2	C0 87	0000		9234	IPG460 BC	*-*,@UCB	RETURN UNLESS FN IS PGCALL		
	13D6	3C 87	13D3		9235	MVI	IPG460+@Q,@UCB	RESTORE RETURN Q CODE		
	13DA	35 10	144C		9236	IPG480 L	IPGCAD,@IAR	BRANCH TO CALLED ROUTINE		

IPGMDL - PAGING MODULE INTERFACES

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	06/09/20	PAGE 161
					9238	*				
					9239	*	IPGRED READS UMPAGE INTO CORE BY SETTING PARMS IN MOVEVM AND CALLIN			
					9240	*				
				13DE	9241	IPGRED EQU	*			ENTRY POINT
13DE	7C	01	A7		9242	MVI	IPGDPL(,@BR),@DGET			MAKE DISK CALL A READ
13E1	5C	00	41 A1		9243	MVC	IPGVRT(,@BR),IPGVPG(1,@BR)			SET VIRTUAL PAGE NUMBER
13E5	7C	80	89		9244	MVI	IPG560+@Q(,@BR),@NOP			MAKE RETURN TO PGCVAD
					9245	*	FALL THRU TO IPGMOV			



## IPGMDL - PAGING MODULE INTERFACES

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 06/09/20 PAGE 162

```

          9247 *
          9248 * IPGMOV - READ/WRITE VIRTUAL PAGE & CALCULATE DISK ADDRESS
          9249 *
13E8 7C 00 A9      13E8 9250 IPGMOV EQU      *          ENTRY POINT
          9251          MVI      IPGDAD(,@BR),*-*      SET VPGNO IN DADDR
          13E9 9252 IPGVRT EQU      IPGMOV+@Q          VPGNO STORAGE LOCATION
13EB 7C 00 A8      9253          MVI      IPGDAD-1(,@BR),0      ZERO FIRST BYTE IF DISK ADDRESS
13EE 5E 01 A9 A9    9254          ALC      IPGDAD(,@BR),IPGDAD(2,@BR)      MULTIPLY BY 4
13F2 5E 01 A9 A9    9255          ALC      IPGDAD(,@BR),IPGDAD(2,@BR)      *
13F6 5E 01 A9 98    9256          ALC      IPGDAD(,@BR),IPGVMS(2,@BR)      ADD IN V.M. DISK ADDR
13FA 7B 01 A9      9257 IPG500 SBF      IPGDAD(,@BR),@DCYL      DISK - REMOVABLE
13FD 5E 01 A9 9A    9258          ALC      IPGDAD(,@BR),IPGB32(@DADDR,@BR)      ADD TRACK DIFFERENTIAL
1401 5F 00 41 9B    9259 IPG520 SLC      IPGVRT(,@BR),IPGB24(1,@BR)      SUBTRACT ONE TRACK
1405 D0 82 7B      9260          BL      IPG540(,@BR)          BRANCH IF FINISHED
1408 5E 01 A9 9A    9261          ALC      IPGDAD(,@BR),IPGB32(@DADDR,@BR)      ADD TRACK DIFFERENTIAL
140C 5F 00 41 9B    9262          SLC      IPGVRT(,@BR),IPGB24(1,@BR)      SUBTRACT ONE TRACK
1410 D0 82 7B      9263          BL      IPG540(,@BR)          BRANCH IF FINISHED
1413 78 01 A9      9264          TBN      IPGDAD(,@BR),@DCYL      IS ADDR ON FIXED DISK NOW ?
1416 D0 10 52      9265          BT      IPG500(,@BR)          YES - CHANGE DISKS, CONTINUE
1419 7A 01 A9      9266          SBN      IPGDAD(,@BR),@DCYL      NO - CHANGE DISKS
141C 5F 01 A9 9F    9267          SLC      IPGDAD(,@BR),IPGHE0(@DADDR,@BR)      RESET ADDRESS RANGE FOR
1420 D0 87 59      9268          B      IPG520(,@BR)          CONTINUE
          9269 *
          9270 * CALCULATE CORE ADDR
          9271 *
1423 7C 20 AB      9272 IPG540 MVI      IPGCA2-1(,@BR),@PGCSZ      SET END OF CORE ADDR
1426 6F 00 AB 00    9273          SLC      IPGCA2-1(,@BR),0(1,@XR)      SUBTRACT PGNO
          9274 *
          9275 * DO READ/WRITE OPERATION
          9276 *
142A C0 87 0025    9277          B      $DISKN          DO I/O
142E 144F          142F 9278          DC      AL2(IPGDPL)          DPL ADDR
          9279 * NEXT JC COULD BE TO USAGE VALUE EFFICIENCY DPL ADDR ROUTINE
1430 D0 00 36      9280 IPG560 BC      IPGRED(,@BR),*-*          FALL DPL ADDR THRU IF READ CA
1433 C0 87 0025    9281          B      $DISKN          WAIT DPL ADDR TILL I/O OP COMP.
1437 1444          1438 9282          DC      AL2(IPGAIT)          DPL ADDRESS
1439 C0 87 1366    9283          B      IPG220          RETURN TO PGCVAD

```

## S/3 BASIC INTERPRETER PAGING MODULE CONSTANTS/WORK AREA

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 06/09/20 PAGE 163

			9285	*		
			9286	*	PAGING MODULE CONSTANTS INTERNAL TO BOTH BASE REGISTER RANGES	
			9287	*		
143D	15EB	143E	9288	IPGUTA	DC	AL2(IPGUVT-1) USAGE VALUE TABLE ADDRESS
143F	06E0	1440	9289	IPGVMS	DC	XL2'06E0' VIRTUAL MEMORY BEGINNING ADDR
1441	0020	1442	9290	IPGB32	DC	XL(@DADDR)'0020' TRACK DIFFERENTIAL
1443	18	1443	9291	IPGB24	DC	IL1'24' SECTORS PER TRACK
1444	FFF5	1445	9292	IPGDXR	DC	AL(@REGL)(65535-IPGNMX) DECR BY PAGE COUNT
1446	00E0	1447	9293	IPGHE0	DC	XL2'00E0' CONSTANT
			9294	*		
			9295	*	PAGING MODULE WORK AREA INTERNAL TO BOTH BASE REGISTER RANGES	
			9296	*		
1448	00	1448	9297		DC	XL1'00' FIRST BYTE OF IPGVAD - CONSTANT
		1449	9298	IPGVPG	EQU	* VIRTUAL PAGE NUMBER
1449		144A	9299	IPGVAD	DS	CL2 VIRTUAL ADDRESS LOCATION
		144B	9300	IPGCPG	EQU	* CORE PAGE NUMBER
144B		144C	9301	IPGCAD	DS	CL2 CORE ADDRESS LOCATION
144D		144E	9302	IPGUVL	DS	CL(IPGLUV) PAGE USAGE VALUE COUNTER
144D			9303		ORG	*-IPGLUV INITIALIZE USAGE VALUE
144D	0001	144E	9304		DC	XL(IPGLUV)'0001' COUNTER TO ZERO
			9305	*	DISK PARAMETER LISTS	
		1444	9306	IPGAIT	EQU	IPGDXR-1 WAIT FUNCTION
144F		144F	9307	IPGDPL	DS	CL1 DISK PARM LIST (DPL)
1450		1451	9308	IPGDAD	DS	CL2 DISK ADDRESS/IPGSLT WORK AREA
1452	01	1452	9309	IPGB01	DC	XL1'01' SECTOR COUNT
1453		1454	9310	IPGCA2	DS	CL2 CORE ADDR
1454			9311		ORG	*-1 INITIALIZE LOW BYTE OF IPGCA2
1454	00	1454	9312		DC	XL1'00' * TO ZERO
		13A8	9313	IPGBAS	EQU	IPG340+@D1 PAGING MODULE BASE

## S/3 BASIC INTERPRETER SELECT CORE PAGE TO BE REPLACED

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 164
		9315		*****	
		9316	*	IPGSLT SELECTS THE CORE PAGE WITH THE LOWEST USAGE VALUE TO BE	*
		9317	*	REPLACED, PROVIDING IT IS NOT LOCKED IN CORE, THE ENTRIES IN IPGUVT	*
		9318	*	AND IPGLRT ARE SET TO ZERO AFTER TESTING, THE CORE PAGE TABLE ENTRY	*
		9319	*	IS SET TO ZERO AFTER TRANSFERRING THE PGNU TO THE NEWLY REFERENCED	*
		9320	*	PAGE, IF NOT READ ONLY CONTROL IS PASSWD TO PGRTRN, OTHERWISE	*
		9321	*	IPGRED IN CALLED TO BRING IN THE NEW PAGE TO THE SPACE AVAILABLE.	*
		9322		*****	
		9323	*		
		9324	*	IPGSLT ENTRY - SET UP TABLE POINTERS AND INITIAL VALUES	
		9325	*		
		1455 9326	IPGSLT EQU *	ENTRY POINT	
1455 34 02 14B5		9327	ST	IPG780+@OP1,@XR	SAVE XR
1459 C2 02 15E1		9328	LA	IPGLRT-1,@XR	POINT TO IPGLRT
145D E2 02 0A		9329	IPG580 LA	IPGNMX(,@XR),@XR	POINT TO LAST ENTRY OF IPGLRT
1460 5C 01 A9 A6		9330	MVC	IPGDAD(,@BR),IPGUVL(IPGLUV,@BR)	SET HIGH USAGE COUNT FOR
1464 7C 0A CB		9331	IPG600 MVI	IPG660+@DD2(,@BR),IPGNMX	INIT DISP TO LAST PAGE
		9332	*		
		9333	*	LOOP TO FIND REPLACEABLE PAGE	
		9334	*		
1467 B8 01 00		9335	IPG620 TBN	0(,@XR),IPGLBT	IS PAGE LOCKED ?
146A E2 02 0A		9336	IPG640 LA	IPGNMX(,@XR),@XR	POINT TO IPGUVT
146D D0 10 D7		9337	BT	IPG700(,@BR)	BRANCH OVER, PAGE LOCKED
1470 6D 01 A9 00		9338	IPG660 CLC	IPGDAD(,@BR),*-(IPGLUV,@XR)	COMPARE USAGE VALUE
1474 D0 82 D7		9339	BL	IPG700(,@BR)	BRANCH OVER, PAGE NOT REPLACED
		9340	*		
		9341	*	SAVE NEW LOW USAGE VALUE OF CURRENT PAGE	
		9342	*		
1477 5C 00 D6 CB		9343	MVC	IPG680+@DD2(,@BR),IPG660+@DD2(1,@BR)	SAVE NEW PGNO
147B 6C 01 A9 00		9344	IPG680 MVC	IPGDAD(,@BR),*-(IPGLUV,@XR)	SET NEW USAGE VALUE
		9345	*		
		9346	*	UPDATE POINTERS AND CONTINUE	
		9347	*		
147F 76 02 9D		9348	IPG700 A	IPGDXR(,@BR),@XR	DECREMENT XR & REPOINT @ IPGLRT
1482 5F 00 CB AA		9349	SLC	IPG660+@DD2(,@BR),IPGB01(1,@BR)	DECREMENT DISP (PGNO)
1486 D0 01 BF		9350	BNE	IPG620(,@BR)	LOOP
		9351	*		
		9352	*	TEST AND RESET READ ONLY BIT - NO FOLLOWING INST, SHOULD MODIFY TEST	
		9353	*	FALSE BIT.	
		9354	*		
1489 5C 00 E7 D6		9355	MVC	IPG720+@D1(1,@BR),IPG680+@DD2(,@BR)	GET PGNO
148D E2 02 00		9356	IPG720 LA	*-(,@XR),@XR	POINT TO IPGLRT ENTRY
1490 B8 02 00		9357	TBN	0(,@XR),IPGRBT	TEST READ ONLY BIT
1493 BB 03 00		9358	SBF	0(,@XR),IPGRBT+IPGLBT	RESET LOCK AND READ ONLY BITS
		9359	*		
		9360	*	FIND AND RESET IPSTBL ENTRY	
		9361	*		
1496 5C 00 FA D6		9362	MVC	IPG760+@Q(,@BR),IPG680+@DD2(1,@BR)	GET PGNO
149A C2 02 14C9		9363	LA	IPGTBL-1,@XR	POINT TO IPGTBL-1
149E E2 02 01		9364	IPG740 LA	1(,@XR),@XR	UPDATE IPGTBL POINTER
14A1 BD 00 00		9365	IPG760 CLI	0(,@XR),*-*	COMPARE ENTRY TO PGNO
14A4 D0 01 F6		9366	BNE	IPG740(,@BR)	LOOP IF NOT EQUAL
		9367	*	XR = IPGTBL+UPGNO OF REPLACED PAGE	
14A7 BC 00 00		9368	MVI	0(,@XR),@ZERO	RESET IPGTBL ENTRY
14AA 74 02 A9		9369	ST	IPGDAD(,@BR),@XR	SAVE XR
14AD 4F 00 A9 1369		9370	SLC	IPGDAD(1,@BR),IPG220+@OP1	CALCULATE VPNO

## S/3 BASIC INTERPRETER SELECT CORE PAGE TO BE REPLACED

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20 PAGE 165
	14B2	C2	02	0000	9371	IPG780	LA *-*,@XR	RESTORE XR
	14B6	9C	00	00 D6	9372		MVC 0(1,@XR),IPG680+@DD2(,@BR)	SET PGNO IN ENTRY FOR REFNCD
	14BA	D0	90	36	9373		BF IPGRED(,@BR)	GO READ NEW PAGE IF OLD RONLY
					9374	*		
					9375	*	FALL THROUGH TO PGRETN IF READ ONLY BIT WAS ON	
					9376	*		
	14BD	7C	87	89	9377		MVI IPG560+@Q(,@BR),@UCB	MAKE RETURN TO IPGRED
	14C0	7C	02	A7	9378		MVI IPGDPL(,@BR),@DPUT	MAKE FUNCTION WRITE
	14C3	5C	00	41 A9	9379		MVC IPGVRT(,@BR),IPGDAD(1,@BR)	SET VPGNO
	14C7	D0	87	40	9380		B IPGMOV(,@BR)	WRITE VIRTUAL PAGE
					9381	*		
					9382	*	IPSTBL - CORE PAGE TABLE - ENTRIES AVAILABLE FOR ALL V.M.	
					9383	*		
	14CA	0000000000000000		14CA	9384	IPGTBL	EQU *	
				15C9	9385		DC XL256'00'	CORE PAGE TABLE
					9386	*		
					9387	*	IPGSTK - STACK FOR BR & ARR SAVE WITH PGCALL & PGRTRN	
					9388	*		
	15CA			15CA	9389	IPGSTK	EQU *	STACK POINTER
				15E1	9390		DS 6XL(@REGL+@REGL)	ROOM FOR 6 LVLS OF PAGE LINKAGE
					9391	*		
					9392	*	IPGLRT - PAGE LOCK AND READ ONLY BIT TABLE - MUST BE DIRECTLY FOLLOW	
					9393	*		BY IPGUVT TABLE
				15E2	9394	IPGLRT	EQU *	TABLE POINTER
	15E2	0000000000000000		15EB	9395		DC XL(IPGNMX)'00'	INITIATE TABLE
					9396	*		
					9397	*	IPGUVT - PAGE USAGE VALUE TABLE - IMMEDIATE FOLLOWS IPGSLT	
					9398	*		
	15EC			15EC	9399	IPGUVT	EQU IPGLRT+IPGNMX	
				15FF	9400		DS XL(IPGNMX*IPGLUV)	ONE FOR EACH CORE PAGE
	15EC				9401		ORG IPGUVT	INITIALIZE
	15EC	0000000000000000		15FF	9402		DC XL(IPGNMX*IPGLUV)'00'	* TO ZERO
					9403	*		
				12EA	9404	IPGSZ1	EQU IPG100	PGSYSZ REFERENCE
				1373	9405	IPGSZ2	EQU IPG240	PGSYSZ REFERENCE
				1423	9406	IPGSZ3	EQU IPG540	PGSYSZ REFERENCE
					9407	*		
				13C4	9408	IPGUT1	EQU IPG400	IPGUVT REFERENCE
				143E	9409	IPGUT2	EQU IPGUTA	IPGUVT REFERENCE
					9410	*		
				1445	9411	IPGMX1	EQU IPGDXR	IPGNMX REFERENCE
				13C4	9412	IPGMX2	EQU IPG400	IPGNMX REFERENCE
				145D	9413	IPGMX3	EQU IPG580	IPGNMX REFERENCE
				146A	9414	IPGMX4	EQU IPG640	IPGNMX REFERENCE
				1464	9415	IPGMX5	EQU IPG600	IPGNMX REFERENCE
					9416	*		
					9417	*****		
					9418	*		
					9419	*	END OF VIRTUAL MEMORY PAGING MODULE CODING	
					9420	*		

## S/3 BASIC INTERPRETER COMMON ADDRESS REFERENCE EQUATES

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 06/09/20 PAGE 166
		9422		*****	
		9423	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		9424	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
		9425	*		*
		9426		*****	
		9428		*****	
		9429	*	CORE RESIDENT ROUTINE ENTRY POINTS AND PARAMETER ADDRESSES	
		9430		*****	
		9431	*		
	0607	9432	IZINIT EQU	IMINIT	ENTRY - INTEPRETFR INITIATOR
	0607	9433	IZFWRK EQU	IMIWRK	INTERPRETER FUNCTION WORK AREA
	0639	9434	IZSTKB EQU	IMISTB	RUN-TIME STACK BASE CORE ADDR
		9435	*		
	0C5C	9436	IZINTR EQU	INTERP	ENTRY - INTERPRETER EXECUTIVE
	0C68	9437	IZNPAG EQU	INTPAG	ENTRY - RESET EXECUTION CONTROL
	0C74	9438	IZXAD4 EQU	INTAD4	ENTRY - INCR IAR, 4-BYTE INST
	0C7B	9439	IZXAD3 EQU	INTAD3	ENTRY - INCR IAR, 3-BYTE INST
	0C82	9440	IZXAD2 EQU	INTAD2	ENTRY - INCR IAR, 2-BYTE INST
	0C89	9441	IZXAD1 EQU	INTAD1	ENTRY - INCR IAR, 1-BYTE INST
	0C9D	9442	IZADST EQU	INTADS	ENTRY - INCR STACK POINTER RTN
	0CAB	9443	IZXERR EQU	INTERR	ENTRY - EXECUTION ERROR RTN
	0C60	9444	IZBASE EQU	INT010	INTERPRETER BASE CORE ADDRESS
	0D4E	9445	IZSTAK EQU	INTSTP	RUN-TIME STACK POINTER
	0D4F	9446	IZSTKI EQU	INTSTI	RUN-TIME STACK POINTER INCR
	0C61	9447	IZXPAG EQU	INTXPG	CURRENT PSEUDO INSTRUCTION PAGE
	0D4C	9448	IZXIAR EQU	INTIAR	PSEUDO INSTRUCTION CORE ADDR
	0CBC	9449	IZERRC EQU	INTERC	EXECUTION ERROR CODE BYTE
	0D53	9450	IZDATA EQU	INTDAT	INTERNAL DATA FILE VADDR PT
	0D55	9451	IZDATI EQU	INTDT1	DATA FILE 1ST ELEMENT VADDR
	0D57	9452	IZPARM EQU	INTPAR	PARAMETER COMMUNICATION AREA
	0D59	9453	IZWRK1 EQU	INTWK1	GENERAL PURPOSE WORK AREA 1
	0D5B	9454	IZWRK2 EQU	INTWK2	GENERAL PURPOSE WORK AREA 2
	0D51	9455	IZSTHA EQU	INTSHA	CURRENT STMT STH INST VADDR
	0CDE	9456	IZIRSW EQU	INTISW	IMAGE STMT REFERENCE SWITCH
	0CE9	9457	IZRESW EQU	INTRSW	STATEMENT RECURSION ERR SWITCH
	0D2B	9458	IZTFSW EQU	INTTSW	INTERPRETER TRACE FLOW SWITCH
	0D3E	9459	IZCBN1 EQU	INTBN1	BINARY CONSTANT, +1
	0D3F	9460	IZCBN2 EQU	INTBN2	BINARY CONSTANT, +2
	0D40	9461	IZCBN3 EQU	INTBN3	BINARY CONSTANT, +3
	0D41	9462	IZCBN4 EQU	INTBN4	BINARY CONSTANT, +4
	0D43	9463	IZCBM1 EQU	INTBM1	BINARY CONSTANT, -1
	0D44	9464	IZCL1F EQU	INTL1F	LENGTH CONSTANT, 1 FLT VALUE
	0D45	9465	IZCL2F EQU	INTL2F	LENGTH CONSTANT, 2 FLT VALUES
	0D46	9466	IZCL1C EQU	INTL1C	LENGTH CONSTANT, 1 CHAR VALUE
	0D47	9467	IZCL2C EQU	INTL2C	LENGTH CONSTANT, 2 CHAR VALUES
	0D49	9468	IZCLVA EQU	INTLVA	LENGTH CONSTANT, VIRTUAL ADDR
	0D4A	9469	IZCLFA EQU	INTLFA	LENGTH CONSTANT, FLT VAL & VADDR
		9470	*		
	0D5C	9471	IZRNSW EQU	INTRND	RANDOM NUMBER INITLZN SWITCH
	0001	9472	IZRNMK EQU	INTRNM	RANDOM NUMBER INITLZN SW MASK
		9473	*		* SW ON = RND FUNC PRIOR USAGE
		9474	*		
	0DC5	9475	IZINDR EQU	INTPIN	PRINT USING INDICATOR BYTE
	0DC6	9476	IZIMLN EQU	INTPIL	IMAGE ASSEMBLY BYTE LENGTH
	0DC8	9477	IZPUB1 EQU	INTPB1	IMAGE BUFFER 1 CORE ADDRESS



## S/3 BASIC INTERPRETER COMMON ADDRESS REFERENCE EQUATES

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 06/09/20 PAGE 167	
		0DCA	9478	IZPUB2	EQU	INTPB2	IMAGE BUFFER 2 CORE ADDRESS
		0DCC	9479	IZIMPT	EQU	INTPIP	IMAGE SCAN POINTER
		0DCE	9480	IZIMC1	EQU	INTPC1	IMAGE CONV SPEC 1ST CHAR PT
		0DD0	9481	IZSDPT	EQU	INTPDP	IMAGE CONV SPEC DECML POINT PT
		0D5A	9482	IZCSCT	EQU	INTPCC	IMAGE CONV SPEC COUNTERS
		0D58	9483	IZSSCT	EQU	INTPSC	IMAGE CONV SPEC DIGIT COUNT
		0D59	9484	IZSDCT	EQU	INTPDC	IMAGE CONV SPEC FRACTION COUNT
		0D5A	9485	IZSFCT	EQU	INTPFC	IMAGE CONV SPEC INTEGER COUNT
		0D5B	9486	IZSICT	EQU	INTPIC	IMAGE CONV SPEC CHAR COUNT
		0D56	9487	IZADUX	EQU	INTPJX	ADJUSTED EXPONENT FOR E-FORMAT
			9488	*			
		0DD1	9489	IZFACT	EQU	INTFAT	FUNCTION ACTIVITY TBL BASE ADDR
		0DE6	9490	IZFATE	EQU	INTFTE	FUNCTION ACTIVITY TBL END ADDR
		0DE8	9491	IZFATP	EQU	INTFAP	FUNCTION ACTIVITY TBL POINTER
			9492	*			
		0D5D	9493	IZSFFO	EQU	INTBAT	FILE EXEC RTNS OVERLAY CADDR
			9494	*			
		119D	9495	IZBSET	EQU	ICBSET	ENTRY - SET BRANCH EXEC ADDR
		117B	9496	IZBRCN	EQU	ICB090+@Q	BRANCH CONDITION STATUS BYTE
			9497	*			
		0B50	9498	IZSTCK	EQU	ISTACK	ENTRY - STACK V.M. ELEMENT
		0BA2	9499	IZSLNG	EQU	ISTLNG	ELEMENT LENGTH INPUT PARAMETER
		0BA1	9500	IZSLLC	EQU	ISTLLC	STACKED ELEMENT LENGTH CODE
			9501	*			
		0BB0	9502	IZUSTK	EQU	IUSTAK	ENTRY - UNSTACK V.M. ELEMENT
		0C3A	9503	IZULNG	EQU	IUSLNG	ELEMENT LENGTH INPUT PARAMETER
		0BC1	9504	IZDMSW	EQU	IUSDSW	SATA TYPE MATCHING SWITCH
			9505	*			
		0A27	9506	IZCPUF	EQU	CPUFLT	ENTRY - FLOATING POINT UNPACKER
			9507	*			
		0A85	9508	IZCUPF	EQU	CUPFLT	ENTRY - FLOATING POINT PACKER
			9509	*			
		0AE3	9510	IZCFBS	EQU	CAFPBS	ENTRY - FLT TO BIN SUBSC CONV
			9511	*			
		075D	9512	IZFADD	EQU	FDIADD	ENTRY - FLOATING ADDITION RTN
		0751	9513	IZFSUB	EQU	FDISUB	ENTRY - FLOATING SUBTRACT RTN
			9514	*			
		082A	9515	IZFMPY	EQU	FZIMPY	ENTRY - FLOATING MULTIPLY RTN
			9516	*			
		0919	9517	IZFDVD	EQU	FFIDVD	ENTRY - FLOATING DIVISION RTN
			9518	*			
		1358	9519	IZCVAD	EQU	IPGCVA	ENTRY - PAGING RTN CONV VADDR
		1349	9520	IZMDFY	EQU	IPGMOD	ENTRY - CONVERT VADDR FOR MODFY
		1354	9521	IZLOCK	EQU	IPGLOK	ENTRY - LOCK AND CONVERT VADDR
		1350	9522	IZUNLK	EQU	IPGULK	ENTRY - UNLOCK A VIRTUAL PAGE
		1329	9523	IZLDBR	EQU	IPGLBR	ENTRY - CONVERT VADDR, LOAD BR
		1330	9524	IZLDXR	EQU	IPGLXR	ENTRY - CONVERT VADDR, LOAD XR
		12B1	9525	IZCALL	EQU	IPGKAL	ENTRY - CALL VIRT MEMORY RTN 1-4
		12D3	9526	IZRTRN	EQU	IPGRTN	ENTRY - RETURN FROM V.M. ROUTINE
		1449	9527	IZPGNO	EQU	IPGVPG	VIRTUAL PAGE INPUT PARAMETER
		144A	9528	IZPGDS	EQU	IPGVAD	VIRTUAL PAGE DISP INPUT PARM
		144A	9529	IZVADR	EQU	IPGVAD	VIRTUAL PAGE INPUT PARM
		144C	9530	IZCADR	EQU	IPGCAD	CORE ADDRESS OUTPUT PARAMETER
		14CA	9531	IZPGTB	EQU	IPGTBL	PAGE REFERENCE TABLE CORE ADDR
		15E2	9532	IZPLRT	EQU	IPGLRT	CORE PAGE STATUS TABLES ADDR
		15CA	9533	IZPSTK	EQU	IPGSTK	PAGE LINKAGE STACK CORE ADDRESS

[illegible][illegible][illegible]



S/3 BASIC INTERPRETER COMMON ADDRESS REFERENCE EQUATES

ERR LOC    OBJECT CODE            ADDR STMT SOURCE STATEMENT                    VER 15, MOD 00    06/09/20    PAGE 169

9537 \*\*\*\*\*  
9538 \* END OF S/3 BASIC INTERPRETER CORE RESIDENT SECTION  
9539 \*\*\*\*\*  
9540 \*  
9541 \* END OF INTERPRETER COMMON SECTION CODING  
9542 \*

FFFF 9543            END

TOTAL STATEMENTS IN ERROR IN THIS ASSEMBLY =        0

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 170

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$\$\$\$\$	001	0600	3277	
\$\$\$\$\$1	040	0750	3455	
\$\$ZERO	001	0000	0409	0410 0412 0413 0414 0418
\$ABORT	001	0010	0522	6014
\$BASIC	001	0080	0580	
\$BIGCD	001	0080	0656	
\$BLDPL	001	0579	0789	0791
\$BLNOE	001	0569	0779	
\$BLOAD	001	0522	0770	0772 0775 0788 0789
\$BLRTN	001	0550	0778	0779
\$BRSAV	001	03C5	0467	0468
\$BSADR	001	0587	0794	0796
\$BUFPT	001	03E3	0675	0676
\$CABLD	001	04B4	0748	0749
\$CAERK	001	0469	0725	0728 5929
\$CAERR	001	03CD	0473	0475 5923*
\$CAIPL	001	049D	0744	0746
\$CALLI	001	0008	0665	
\$CARDI	001	0001	0436	
\$CARPL	001	04A1	0746	0748 6272
\$CIENT	001	0483	0735	0736
\$CIEXT	001	0480	0734	0735
\$CIMSK	001	0476	0731	0734 3386* 5995*
\$CISUS	001	0496	0739	0744
\$CLBFR	001	0010	0623	
\$CMDKY	001	0008	0535	
\$CMODE	001	0002	0585	
\$CONFIG	001	03DD	0648	0658
\$CRPOS	001	03E2	0674	0675
\$CRTAD	001	044D	0713	0714
\$CRTAV	001	0002	0529	
\$CRTDN	001	0002	0553	
\$CRTIN	001	03D3	0550	0557
\$CRTNO	001	0004	0532	
\$CRTPU	001	0004	0554	
\$CRTSP	001	0008	0555	
\$CRTUP	001	0001	0552	
\$CRUSH	001	0080	0661	
\$CSDPL	001	050E	0760	0761
\$C0001	001	0464	0717	0723 9142 9144 9226
\$DATE	001	043A	0698	0699
\$DBGUF	001	03E0	0660	0669
\$DBLOK	001	0001	0610	
\$DFDET	001	03E8	0681	0682
\$DISKN	001	0025	0412	3313 3351 3381 9277 9281
\$DKERR	001	0008	0591	
\$DKSIZ	001	03D7	0635	0643 0684
\$DK100	001	0001	0637	
\$DK200	001	0002	0638	
\$DK400	001	0004	0639	
\$DK600	001	0008	0640	
\$DK800	001	0010	0641	
\$DPLSV	001	0449	0709	0711
\$DTNMB	001	0040	0456	
\$DTRDR	001	0040	0544	
\$ENDNU	001	0600	0803	1437 8317

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 171

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$ERDPL	001	046F	0728	0730
\$ERFIL	001	0040	0483	
\$ERHRD	001	0004	0615	
\$ERKEY	001	0080	0487	
\$ERLOG	001	0345	0417	
\$ERMAD	001	0472	0730	0731
\$ERPND	001	0004	0588	
\$ERRCT	001	03CF	0489	
\$ERRPG	001	03CE	0477	
\$ERSFL	001	0035	0482	
\$ERSTK	001	0030	0480	
\$ER050	001	0363	0418	
\$ER1N2	001	0050	0485	
\$EXADR	001	0517	0763	0765
\$EXCMD	001	0001	0517	3388 5928 6271
\$EXFTR	001	043B	0699	0704 3295 3306 3318 3325 3326 3327
\$FCIND	001	0010	0595	
\$FDIND	001	0040	0602	
\$FEARR	001	0004	0410	
\$FEMAP	001	0588	0796	0797
\$FILIB	001	03DA	0646	0647
\$FITIN	001	0010	0571	
\$FUIND	001	0020	0600	
\$GUFIO	001	0583	0793	0794
\$GUFIR	001	0008	0445	
\$HISTE	001	042E	0696	0697
\$HIST1	001	0435	0697	0698
\$HRDER	001	0020	0541	
\$INDR1	001	03D4	0557	0583
\$INDR2	001	03D5	0583	0608
\$INDR3	001	03D6	0608	0635
\$INLNO	001	03CF	0475	0477 0489 0496 3375 3376 3377* 5974 6023*
\$INRPT	001	0020	0453	
\$IOIND	001	03D2	0524	0550
\$IOPGS	001	0010	0664	
\$IOYES	001	0002	0439	
\$IPLDV	001	05FF	0800	0803
\$IRKEY	001	0020	0663	
\$KEYBD	001	03E1	0669	0674
\$KEYCD	001	03C3	0433	0467
\$KEYDT	001	0040	0577	
\$KE090	001	00DE	0413	
\$KE130	001	01D5	0414	
\$KYBSY	001	0010	0450	
\$LDRTN	001	0571	0788	
\$LEVEL	001	03DF	0658	0660
\$LIST	001	0002	0612	
\$LMRGN	001	03C1	0428	0430
\$LNPTR	001	0080	0547	
\$LOADB	001	054A	0772	
\$LOADR	001	051A	0765	0768
\$LPRIO	001	03EA	0682	
\$LPROS	001	03E5	0677	0679
\$LPRP3	001	03E4	0676	0677
\$MOUNT	001	0020	0626	
\$MPDWN	001	0001	0526	

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 172

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$NEXTB	001	03E6	0679	0680
\$NEXTL	001	03E7	0680	0681
\$NOENB	001	0008	0618	
\$NOLST	001	0004	0442	
\$NUCBS	001	03C0	0425	0426
\$NWRKF	001	0080	0631	
\$NWRKR	001	0040	0628	
\$PASWD	001	042D	0695	0696
\$PAUSD	001	04BA	0749	0751 6006 6224
\$PAUSE	001	0002	0519	5996 6004 6019 6223
\$PGMDT	001	0020	0574	
\$PGMST	001	0010	0538	
\$PKERT	001	0419	0693	0695
\$PLST1	001	0454	0714	0715
\$PLST2	001	045B	0715	0716
\$PLST3	001	0462	0716	0717
\$PRDEV	001	044B	0711	0713
\$PRESN	001	0002	0562	
\$PROCI	001	0001	0559	
\$PRPOS	001	03C2	0430	0433
\$PSDBR	001	04FA	0754	
\$PSDXR	001	04F2	0753	0754
\$PSTEP	001	0004	0520	6004 6019
\$PSTMT	001	0008	0521	6019 6223
\$PTCH1	001	03F5	0684	0688
\$READY	001	0080	0604	
\$REORD	001	0040	0662	
\$RLOAD	001	051E	0768	0770
\$RMGRN	001	03C0	0426	0428
\$RSTR	001	04D6	0751	0753 0755 0760
\$RUNIT	001	0001	0498	
\$SFAID	001	050D	0756	
\$SPRNT	001	0465	0723	0725
\$SRTRN	001	04FE	0755	0756
\$STEPT	001	0002	0499	5999
\$SWPCR	001	0511	0761	0763
\$TABLN	001	03CB	0470	0473
\$TFLOW	001	0008	0505	3366
\$TRACE	001	0004	0500	5346 6034
\$TRALL	001	0010	0506	
\$TROVR	001	054E	0775	0778
\$TRUNK	001	0080	0458	
\$TRVAR	001	0020	0507	
\$UNMSK	001	048D	0736	0739 5994
\$USRDR	001	03DC	0647	0648
\$VMDEF	001	0080	0511	3387
\$VOLF1	001	03FE	0690	0691
\$VOLF2	001	040E	0692	
\$VOLID	001	03F6	0688	0689 0693
\$VOLR1	001	03F6	0689	0690
\$VOLR2	001	0406	0691	0692
\$WAITF	001	057F	0791	0793 3382
\$WFDEF	001	0040	0705	
\$WFLOK	001	0008	0568	
\$WFNME	001	0443	0704	0709
\$WSIND	001	0004	0565	

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 173

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$XIND1	001	03D0	0496	0515 3366 3387* 5346 5999 6034
\$XIND2	001	03D1	0515	0524 3388* 5928* 5996 6004* 6014 6019* 6223* 6271*
\$XIND3	001	03D8	0643	0646
\$XPREC	001	0040	0508	
\$XRSAV	001	03C7	0468	0470
\$ZTRAD	001	05A2	0797	
\$12K	001	0004	0652	
\$16CKY	001	0008	0654	
\$16K	001	0002	0651	
\$22IMP	001	0001	0649	
\$\$\$INS	001	0600	3274	3276
#\$@INS	001	0010	3275	
#\$INST	001	0020	3273	
#INST	001	0607	3280	
#INSTD	001	0000	0001	
@@E001	001	0000	1341	1343
@@E003	001	0001	1343	1345
@@E004	001	0002	1345	1347
@@E005	001	0003	1347	1349
@@E006	001	0004	1349	1351
@@E007	001	0005	1351	1353
@@E008	001	0006	1353	1355
@@E009	001	0007	1355	1357
@@E010	001	0008	1357	1359
@@E011	001	0009	1359	1361
@@E012	001	000A	1361	1363
@@E013	001	000B	1363	1365
@@E014	001	000C	1365	1367
@@E015	001	000D	1367	1369
@@E016	001	000E	1369	1371
@@E017	001	000F	1371	1373
@@E018	001	0010	1373	1375
@@E019	001	0011	1375	1377
@@E020	001	0012	1377	1379
@@E021	001	0013	1379	1381
@@E023	001	0014	1381	1383
@@E024	001	0015	1383	1385
@@E025	001	0016	1385	1387
@@E026	001	0017	1387	1389
@@E027	001	0018	1389	1391
@@E028	001	0019	1391	1393
@@E029	001	001A	1393	1395
@@E030	001	001B	1395	1397
@@E031	001	001C	1397	1399
@@E032	001	001D	1399	1401
@@E035	001	001E	1401	1403
@@E036	001	001F	1403	1405
@@E037	001	0020	1405	1407
@@E038	001	0021	1407	1409
@@E039	001	0022	1409	1411
@@E040	001	0023	1411	1413
@@E041	001	0024	1413	1415
@@E042	001	0025	1415	1417
@@E043	001	0026	1417	1419
@@E044	001	0027	1419	1421
@@E045	001	0028	1421	1423

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 174

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E046	001	0029	1423	1425
@@E060	001	002A	1425	1427
@@E080	001	002B	1427	
@@E100	001	0000	0813	0815
@@E101	001	0001	0815	0817
@@E102	001	0002	0817	0819
@@E103	001	0003	0819	0821
@@E110	001	0004	0821	0823
@@E112	001	0005	0823	0825
@@E113	001	0006	0825	0827
@@E114	001	0007	0827	0829
@@E115	001	0008	0829	0831
@@E116	001	0009	0831	0833
@@E117	001	000A	0833	0835
@@E120	001	000B	0835	0837
@@E122	001	000C	0837	0839
@@E123	001	000D	0839	0841
@@E124	001	000E	0841	0843
@@E129	001	000F	0843	0845
@@E130	001	0010	0845	0847
@@E131	001	0011	0847	0849
@@E133	001	0012	0849	0851
@@E134	001	0013	0851	0853
@@E135	001	0014	0853	0855
@@E136	001	0015	0855	0857
@@E137	001	0016	0857	0859
@@E138	001	0017	0859	0861
@@E139	001	0018	0861	0863
@@E142	001	0019	0863	0865
@@E143	001	001A	0865	0867
@@E150	001	001B	0867	0869
@@E151	001	001C	0869	0871
@@E160	001	001D	0871	0873
@@E162	001	001E	0873	0875
@@E163	001	001F	0875	0877
@@E164	001	0020	0877	0879
@@E200	001	0021	0879	0881
@@E205	001	0022	0881	0883
@@E210	001	0023	0883	0885
@@E211	001	0024	0885	0887
@@E212	001	0025	0887	0889
@@E213	001	0026	0889	0891
@@E215	001	0027	0891	0893
@@E216	001	0028	0893	0895
@@E217	001	0029	0895	0897
@@E220	001	002A	0897	0899
@@E221	001	002B	0899	0901
@@E222	001	002C	0901	0903
@@E223	001	002D	0903	0905
@@E225	001	002E	0905	0907
@@E226	001	002F	0907	0909
@@E227	001	0030	0909	0911
@@E228	001	0031	0911	0913
@@E229	001	0032	0913	0915
@@E230	001	0033	0915	0917
@@E232	001	0034	0917	0919

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 175

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E234	001	0035	0919	0921
@@E237	001	0036	0921	0923
@@E240	001	0037	0923	0925
@@E241	001	0038	0925	0927 3035
@@E242	001	0039	0927	0929
@@E248	001	003A	0929	0931
@@E249	001	003B	0931	0933
@@E250	001	003C	0933	0935
@@E251	001	003D	0935	0937
@@E252	001	003E	0937	0939
@@E253	001	003F	0939	0941
@@E254	001	0040	0941	0943
@@E255	001	0041	0943	0945
@@E256	001	0042	0945	0947
@@E300	001	0043	0947	0949
@@E301	001	0044	0949	0951
@@E302	001	0045	0951	0953
@@E303	001	0046	0953	0955
@@E304	001	0047	0955	0957
@@E305	001	0048	0957	0959
@@E308	001	0049	0959	0961
@@E310	001	004A	0961	0963
@@E315	001	004B	0963	0965
@@E316	001	004C	0965	0967
@@E320	001	004D	0967	0969
@@E325	001	004E	0969	0971
@@E330	001	004F	0971	0973
@@E335	001	0050	0973	0975
@@E338	001	0051	0975	0977
@@E340	001	0052	0977	0979
@@E350	001	0053	0979	0981
@@E351	001	0054	0981	0983
@@E352	001	0055	0983	0985
@@E360	001	0056	0985	0987
@@E361	001	0057	0987	0989
@@E362	001	0058	0989	0991
@@E371	001	0059	0991	0993
@@E380	001	005A	0993	0995
@@E390	001	005B	0995	0997
@@E400	001	005C	0997	0999
@@E410	001	005D	0999	1001
@@E415	001	005E	1001	1003
@@E417	001	005F	1003	1005
@@E420	001	0060	1005	1007
@@E430	001	0061	1007	1009
@@E432	001	0062	1009	1011
@@E433	001	0063	1011	1013
@@E450	001	0064	1013	1015
@@E451	001	0065	1015	1017
@@E460	001	0066	1017	1019
@@E461	001	0067	1019	1021
@@E464	001	0068	1021	1023
@@E465	001	0069	1023	1025
@@E466	001	006A	1025	1027
@@E467	001	006B	1027	1029
@@E469	001	006C	1029	1031



## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 176

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E470	001	006D	1031	1033
@@E471	001	006E	1033	1035
@@E473	001	006F	1035	1037
@@E474	001	0070	1037	1039
@@E475	001	0071	1039	1041
@@E476	001	0072	1041	1043
@@E477	001	0073	1043	1045
@@E478	001	0074	1045	1047
@@E479	001	0075	1047	1049
@@E480	001	0076	1049	1051
@@E481	001	0077	1051	1053
@@E482	001	0078	1053	1055
@@E483	001	0079	1055	1057
@@E484	001	007A	1057	1059
@@E485	001	007B	1059	1061
@@E486	001	007C	1061	1063
@@E487	001	007D	1063	1065
@@E488	001	007E	1065	1067
@@E489	001	007F	1067	1069
@@E490	001	0080	1069	1071
@@E491	001	0081	1071	1073
@@E492	001	0082	1073	1075
@@E493	001	0083	1075	1077
@@E494	001	0084	1077	1079
@@E495	001	0085	1079	1081
@@E496	001	0086	1081	1083
@@E497	001	0087	1083	1085
@@E498	001	0088	1085	1087
@@E500	001	0089	1087	1089
@@E501	001	008A	1089	1091
@@E530	001	008B	1091	1093
@@E531	001	008C	1093	1095
@@E535	001	008D	1095	1097
@@E540	001	008E	1097	1099
@@E541	001	008F	1099	1101
@@E542	001	0090	1101	1103
@@E543	001	0091	1103	1105
@@E544	001	0092	1105	1107
@@E545	001	0093	1107	1109
@@E546	001	0094	1109	1111
@@E547	001	0095	1111	1113
@@E548	001	FFFF	1317	
@@E549	001	0096	1113	1115
@@E550	001	0097	1115	1117
@@E551	001	0098	1117	1119
@@E552	001	0099	1119	1121
@@E553	001	009A	1121	1123
@@E554	001	009B	1123	1125
@@E555	001	009C	1125	1127
@@E556	001	009D	1127	1129
@@E558	001	009E	1129	1131
@@E570	001	009F	1131	1133
@@E571	001	00A0	1133	1135
@@E572	001	00A1	1135	1137
@@E573	001	00A2	1137	1139
@@E574	001	00A3	1139	1141

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 177

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E575	001	FFFF	1319	
@@E578	001	00A4	1141	1143
@@E579	001	FFFF	1321	
@@E580	001	FFFF	1323	
@@E585	001	00A5	1143	1145
@@E595	001	FFFF	1325	
@@E597	001	FFFF	1327	
@@E598	001	FFFF	1329	
@@E600	001	00A6	1145	1147
@@E601	001	00A7	1147	1149
@@E602	001	00A8	1149	1151
@@E603	001	00A9	1151	1153
@@E604	001	00AA	1153	1155
@@E606	001	00AB	1155	1157
@@E607	001	00AC	1157	1159
@@E608	001	00AD	1159	1161
@@E609	001	00AE	1161	1163
@@E610	001	00AF	1163	1165
@@E611	001	00B0	1165	1167
@@E612	001	00B1	1167	1169
@@E613	001	00B2	1169	1171
@@E614	001	00B3	1171	1173
@@E700	001	00B4	1173	1175 8380
@@E701	001	00B5	1175	1177
@@E710	001	00B6	1177	1179
@@E712	001	00B7	1179	1181
@@E713	001	00B8	1181	1183
@@E714	001	00B9	1183	1185
@@E715	001	00BA	1185	1187
@@E716	001	00BB	1187	1189
@@E717	001	00BC	1189	1191
@@E718	001	00BD	1191	1193
@@E720	001	00BE	1193	1195
@@E721	001	00BF	1195	1197
@@E723	001	00C0	1197	1199 8688
@@E724	001	00C1	1199	1201
@@E725	001	00C2	1201	1203 5962
@@E726	001	00C3	1203	1205 8322
@@E727	001	00C4	1205	1207 5250
@@E728	001	00C5	1207	1209
@@E729	001	00C6	1209	1211 5973
@@E730	001	00C7	1211	1213 5900
@@E732	001	00C8	1213	1215
@@E752	001	00C9	1215	1217
@@E753	001	00CA	1217	1219
@@E754	001	00CB	1219	1221
@@E755	001	00CC	1221	1223
@@E756	001	00CD	1223	1225
@@E757	001	00CE	1225	1227
@@E758	001	00CF	1227	1229
@@E759	001	00D0	1229	1231
@@E760	001	00D1	1231	1233 4797 7718
@@E761	001	00D2	1233	1235
@@E762	001	00D3	1235	1237
@@E763	001	00D4	1237	1239
@@E764	001	00D5	1239	1241

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 178

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E765	001	00D6	1241	1243
@@E766	001	00D7	1243	1245
@@E767	001	00D8	1245	1247
@@E768	001	00D9	1247	1249
@@E769	001	00DA	1249	1251
@@E770	001	00DB	1251	1253
@@E771	001	00DC	1253	1255
@@E772	001	00DD	1255	1257
@@E773	001	00DE	1257	1259
@@E774	001	00DF	1259	1261
@@E775	001	00E0	1261	1263
@@E776	001	00E1	1263	1265
@@E777	001	00E2	1265	1267
@@E778	001	00E3	1267	1269
@@E779	001	00E4	1269	1271
@@E780	001	00E5	1271	1273
@@E781	001	00E6	1273	1275
@@E782	001	00E7	1275	1277
@@E783	001	00E8	1277	1279
@@E784	001	00E9	1279	1281
@@E785	001	00EA	1281	1283
@@E786	001	00EB	1283	1285
@@E790	001	00EC	1285	1287 4073
@@E791	001	00ED	1287	1289 3687 3924 4172
@@E792	001	00EE	1289	1291 3690 3927 4177
@@E793	001	00EF	1291	1293
@@E794	001	00F0	1293	1295
@@E795	001	00F1	1295	1297
@@E796	001	00F2	1297	1299
@@E797	001	00F3	1299	1301
@@E798	001	00F4	1301	1303
@@E800	001	FFFF	1331	
@@E801	001	FFFF	1333	
@@E802	001	FFFF	1335	
@@E803	001	FFFF	1337	
@@E804	001	FFFF	1339	
@@E900	001	00F5	1303	1305 3031
@@E901	001	00F6	1305	1307 3033
@@E902	001	00F7	1307	1309 3032
@@E903	001	00F8	1309	1311 3034
@@E905	001	00F9	1311	1313
@@E906	001	00FA	1313	1315
@@E910	001	00FB	1315	3030
@ALTFL	001	0001	0251	
@ARR	001	0008	0016	3581 3833 4067 4328 4542 4777 4972 5222 5890 6278 7564 7646 7703 9082 9092* 9102 9121 9142* 9143 9144* 9177
@ASIGN	001	007C	0071	
@ASTER	001	005C	0069	
@BCRDL	001	0050	0088	
@BE	001	0081	0043	5978 5982
@BF	001	0090	0052	5241
@BH	001	0084	0041	7952
@BKSPC	001	0010	0348	
@BL	001	0082	0042	7948
@BLANK	001	0040	0065	
@BM	001	0082	0054	

CROSS REFERENCE																		
SYMBOL	LEN	VALUE	DEFN	REFERENCES												VER 15, MOD 00	06/09/20	PAGE 179
@BNE	001	0001	0046															
@BNH	001	0004	0044															
@BNL	001	0002	0045															
@BNM	001	0002	0057															
@BNOL	001	0020	0050															
@BNOZ	001	0008	0049															
@BNP	001	0004	0056															
@BNZ	001	0001	0058															
@BOL	001	00A0	0048															
@BOZ	001	0088	0047															
@BP	001	0084	0053															
@BR	001	0001	0013	3290	3291*	3296	3306	3307	3307	3308	3308	3318	3319	3319	3320			
				3320	3321	3321	3322	3322	3323	3323	3329	3330	3331	3333	3334			
				3335	3336	3337	3357	3357	3361	3361	3362	3377	3566	3579	3580*			
				3581	3588	3588	3589	3599	3600	3604	3611	3620	3621	3622	3630			
				3630	3643	3655	3657	3659	3659	3660	3661	3669	3676	3676	3680			
				3686	3689	3691*	3817	3831	3832*	3833	3838	3848	3849	3852	3864			
				3865	3865	3870	3871	3872	3873	3874	3875	3876	3877	3878	3879			
				3880	3882	3883	3883	3884	3894	3896	3898	3900	3906	3909	3913			
				3914	3919	3920	3921	3921	3922	3922	3923	3925	3925	3926	3928			
				3928	3929	3930*	4061	4065	4066*	4067	4073	4074	4079	4080	4081			
				4095	4096	4111	4111	4112	4112	4118	4119	4120	4121	4122	4123			
				4124	4125	4129	4130	4131	4135	4136	4137	4142	4142	4143	4155			
				4156	4158	4158	4159	4160	4162	4163	4169	4170	4170	4171	4171			
				4172	4173	4174	4174	4176	4177	4179	4179	4180	4181*	4774	4775			
				4776*	4777	4791	4798	4802	4803	4804	4804	4809	4809	4810	4816			
				4816	4817	4818	4822	4822	4826	4826	4827	4831	4835*	4969	4970			
				4971*	4972	4977	4977	4982	4983	4983	4997	4998	4999	4999	5004*			
				5005	5006*	5010	5011	5011	5013	5026*	5028	5038*	5220	5221	5232*			
				5242	5259	5260*	5268	5277	5277	5282	5283	5283	5294	5295	5296			
				5296	5301*	5302	5303*	5307	5308	5308	5310	5324*	5326	5336*	5783			
				5784*	5810	5823	5823	5836	5836	5849	5849	5862	5862	5874	5875			
				5876	5890	5894	5894	5900	5939	5940	5942	5954	5955	5956	5956			
				5962	5964	5973	5976	5982	5983	6010	6027	6035	6042	6228	6230			
				6242	6247	6247	6512	6519	6519	6530	6530	6532	6579	6579	6584			
				6585	6589	6589	6590	6599	6608	6609	6609	6613	6621	6622	6626			
				6627	6629	6821	6828	6828	6836	6843	6850	6860	6870	6871	6875			
				7098														

[illegible]

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 181

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@DBFR2	001	0005	0130	
@DBUSY	001	0002	0245	
@DCALK	001	0001	0081	
@DCBCY	001	0009	0115	2015
@DCBT1	001	0050	0117	2018
@DCFLN	001	0004	0229	
@DCNT	001	0003	0128	
@DCRID	001	0001	0243	
@DCST1	001	0040	0116	2016
@DCTRL	001	0000	0125	
@DCTRW	001	0000	0242	
@DCWID	001	0001	0239	
@DCYL	001	0001	0126	9257 9264 9266
@DCYMV	001	0001	0230	
@DD2	001	0003	0030	3588 3588* 3620* 3630 3864* 4802* 4977 4977* 4983 4997* 4998* 4999 4999* 5010* 5011 5029 5033* 5053 5277 5277* 5283 5294* 5295* 5296 5296* 5327 5331* 5368 8013* 9331* 9343 9343* 9349* 9355 9362 9372
@DEFLG	001	0002	0252	
@DERCE	001	0020	0282	
@DERD2	001	0008	0274	
@DEREQ	001	0010	0273	
@DERIN	001	0040	0271	
@DERMA	001	0020	0272	
@DERNR	001	0004	0275	
@DERR	001	0000	0246	
@DERSC	001	0001	0277	
@DERTC	001	0002	0276	
@DFCR	001	0006	0232	
@DFDR	001	0004	0233	
@DGET	001	0001	0134	3422 9242
@DHARD	001	0000	0260	
@DLNCT	001	000F	0346	
@DLNLG	001	0040	0345	
@DOLAR	001	005B	0068	
@DOP2	001	0004	0028	
@DPLNG	001	0006	0132	
@DPOS	001	0000	0133	
@DPUT	001	0002	0135	9378
@DREAD	001	0001	0236	
@DSAD	001	0002	0127	
@DSBCY	001	0004	0106	1953
@DSBSY	001	0092	0341	
@DSCS1	001	0000	0107	1954
@DSEEK	001	0000	0235	
@DSIVF	001	0003	0138	
@DSPIN	001	0002	0131	
@DTRSZ	001	0018	0085	
@DUNSF	001	0080	0278	
@DVBCY	001	0007	0108	2012 3423
@DVERY	001	0003	0241	
@DVRFY	001	0031	0136	
@DVST1	001	0002	0247	
@DVST2	001	0003	0248	
@DWAIT	001	00FF	0137	
@DWBCY	001	0005	0103	2009
@DWBIT	001	0002	0237	

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 182

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@DWSIZ	001	00C0	0105	
@DWTB1	001	0003	0104	2010
@DZERO	001	00F0	0064	3632 3633 3850 3895 3905 4072 4078 4086
@D1	001	0002	0026	3335* 3336* 3630* 3865* 4803* 4826* 5055 5307* 5308 5875* 9104* 9105* 9118 9160 9198* 9207* 9214* 9220 9313 9355*
@EOF	001	001C	0077	
@EOFTC	001	0075	0162	
@EOS	001	001E	0076	2025
@ER37B	001	00F0	0362	
@FDDBC	001	0000	0195	
@FDE1	001	000C	0200	
@FDFNA	001	000B	0198	
@FDHLN	001	0002	0208	
@FDLNC	001	0002	0193	
@FDNSC	001	0003	0210	
@FDSD	001	0000	0206	
@FLACE	001	0009	0197	
@FLDBC	001	0001	0196	
@FLDIN	001	0012	0334	
@FLENT	001	0004	0201	
@FLFNA	001	0002	0199	
@FLHLN	001	0002	0209	
@FLLNC	001	0002	0194	
@FLNSC	001	0001	0211	
@FLSD	001	0001	0207	
@HDRLN	001	0007	0092	
@HSTAD	001	0009	0258	
@HSTEN	001	0007	0257	
@HSTPE	001	0006	0256	
@HSTQR	001	0001	0254	
@HSTSN	001	0005	0255	
@HSTVI	001	000F	0259	
@IAR	001	0010	0017	9236*
@ID37B	001	0040	0398	
@INDEX	001	0001	0156	0157
@INST3	001	0003	0032	5237 5341 5967 5979 5990 6030 8361 9086
@INST4	001	0004	0033	4977 4999 5031 5277 5296 5329 5791 5926
@INST5	001	0005	0034	
@INST6	001	0006	0035	
@IP37B	001	00C0	0397	
@I1IAR	001	00C0	0020	
@KCMDK	001	0020	0308	
@KELOK	001	001B	0307	
@KENAB	001	001E	0305	
@KEXIT	001	001F	0306	
@KEYBD	001	0010	0325	
@KFUNK	001	0010	0328	
@KHARD	001	0011	0333	
@KLEAR	001	000D	0329	
@LINSZ	001	00F4	0084	
@LO37B	001	00F0	0366	
@MAPEN	001	0005	0089	
@MINCR	001	2000	0083	
@MINUS	001	0060	0080	
@NOP	001	0080	0040	2372 3371 3386 4124 5268 5939 5940 5966 5995 6010 7151 7560 8273 8769 8994 9125 9132 9138 9244



## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 183

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@NORFL	001	0000	0253	
@NTRDY	001	00A0	0390	
@NUMBR	001	007B	0070	
@OPD2	001	0004	0029	
@OP1	001	0003	0027	3329* 3579* 3581* 3831* 3833* 4065* 4067* 4328* 4542* 4775* 4777* 4970* 4972* 5221* 5222* 5890* 6278* 7196* 7564* 7646* 7703* 9099* 9100 9101* 9106* 9120 9123* 9175* 9177* 9178* 9197* 9201* 9206* 9208 9208* 9209* 9327* 9370
@OP2	001	0005	0031	3330* 9082* 9092 9143*
@OVRUN	001	0004	0283	
@PBUSY	001	00E2	0295	9087
@PCAR	001	00E6	0292	
@PCNT	001	0003	0227	
@PCTRL	001	0000	0149	
@PCYL	001	0001	0225	
@PC37B	001	00F2	0382	
@PDAR	001	00E4	0291	
@PDATA	001	0003	0151	
@PD37B	001	0080	0396	
@PERR	001	00E0	0298	
@PFLAG	001	0000	0224	
@PFORM	001	00E1	0296	
@PGCSZ	001	0020	0082	0083 9104 9190 9272
@PLITE	001	00E2	0297	
@PLNGH	001	0004	0288	
@PMGCK	001	0020	0299	
@PN37B	001	00F0	0381	
@PPLNG	001	0004	0148	
@PRCNT	001	0001	0150	
@PRETR	001	00C0	0154	
@PRINT	001	0040	0152	0154
@PRITY	001	0080	0332	
@PSAD	001	0002	0226	
@PSIOQ	001	00E0	0294	
@PSIOR	001	0000	0293	
@PSNSQ	001	00E2	0300	
@PSR	001	0004	0015	3848 3913* 4095 4162*
@PWAIT	001	00FF	0158	
@P1IAR	001	0020	0018	
@P2IAR	001	0040	0019	
@Q	001	0001	0024	3325* 3326* 3327* 3334* 3337* 3357* 3361* 3589* 3604* 3621* 3622* 3676* 4073* 4121* 4124* 4172* 4177* 4982* 4983* 5011* 5235 5241* 5244* 5268* 5282* 5283* 5308* 5339 5345* 5371 5789 5924 5939 5940* 5965 5977 5982* 5988 6010* 6028 6199 6200 6202 6205 6208 6228* 7151* 7161* 7554* 7560* 7948* 7952* 8273* 8359 8365* 8730* 8731* 9084 9090* 9124* 9125* 9132* 9138* 9151* 9158 9167* 9199* 9202* 9218 9222* 9235* 9244* 9252 9362* 9377* 9496
@RD37B	001	00F1	0376	
@REGL	001	0002	0012	3333 4203 9112 9292 9390 9390
@RETRN	001	0080	0153	0154
@RLDWN	001	004F	0159	
@RTCNT	001	0003	0290	
@RTRNC	001	0080	0161	
@RT37B	001	0005	0389	
@SBLN	001	0005	0170	
@SBLNL	001	0002	0184	

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 184

@SCTSZ	001	0100	0100												
@SDFLN	001	0007	0090												
@SDF0	001	0000	0166												
@SDF1	001	0001	0167												
@SDF2	001	0002	0168												
@SDF3	001	0003	0169												
@SECCY	001	0030	0086												
@SIST	001	0001	0181												
@SKCTL	001	0000	0240												
@SLASH	001	0061	0067												
@SLAST	001	0002	0183												
@SMIDL	001	0003	0182												
@SNSB0	001	0000	0264												
@SNSB1	001	0001	0265												
@SNSB2	001	0002	0266												
@SNSB3	001	0003	0267												
@SNULL	001	0080	0173												
@SN37B	001	00F2	0370												
@SONLY	001	0000	0180												
@SPINA	001	00A0	0249												
@SPINB	001	00B0	0250												
@STEXT	001	0007	0172												
@STYPE	001	0006	0171												
@SYCNT	001	0002	0289												
@SYLVL	001	0005	3066												
@TBCNT	001	0000	0160												
@TBLEF	001	0010	0155	0157											
@TBLIX	001	0011	0157												
@TJ37B	001	0040	0387												
@TYPAM	001	0002	0331												
@TYPO	001	001C	0330												
@UCB	001	0087	0039	2349	4121	5236	5340	5345	5989	6029	6228	7161	7554	8279	8360
				8365	8997	9085	9090	9196	9199	9200	9202	9234	9235	9377	
@UPARW	001	005A	0078	3049											
@VADDR	001	0002	0141	1746	2182	2194	2195	2196	2196	2210	2213	2215	2239	2240	2241
				2279	2282	2285	2288	2291	2294	2297	2306	2309	2312	2315	2318
				2985	3011	3375	3376	4982	4998	5013	5267	5282	5295	5310	5352
				5790	5914	5919	6040	6060	6061	6075	6077	6078	6183	6184	6242
				6247	6262	6267	6280	6522	6617	6864	7202	7229	7233	7499	7519
				7610	7611	7612	7616	7618	7619	7672	7673	7674	7675	7676	7677
				7678	7679	7918	7941	7990	8010	8011	8017	8311	8313	8371	8409
				8660	8694	8750	8760	8778	8944	8971	8996				
@VENTA	001	0056	0113	2013	2268	5790	8375								
@VMDDV	001	00FE	0114												
@VMFD1	001	0000	0109												
@VMFD2	001	0001	0110												
@VMRS3	001	0002	0112												
@VMTRL	001	0001	0111												
@VOLID	001	0006	0091												
@VQ	001	0001	0025	3639	3639	5005	5028	5302	5326						
@WA37B	001	00FF	0395												
@WSFIT	001	0500	0101												
@WSTBL	001	0503	0102												
@XR	001	0002	0014	3356*	3359	3360	3360*	3570*	3574	3575	3582*	3589	3590	3590	3601
				3602	3602	3603	3604	3605	3605	3606	3620	3621	3622	3631	3631
				3632	3633	3639	3639	3643	3644	3645	3645	3656	3658	3658*	3667*

CROSS REFERENCE													
SYMBOL	LEN	VALUE	DEFN	REFERENCES	VER 15, MOD 00 06/09/20 PAGE 185								
				3668 3677 3678* 3679 3680 3681 3685 3688 3709 3709 3834* 3838									
				3846 3846 3847 3847 3850 3851 3851 3864 3870 3871 3872 3873									
				3874 3876 3877 3878 3879 3880 3882* 3894 3895 3897 3897 3899									
				3899 3900 3901* 3905 3907 3908 3908 3915 3920 3929 4068* 4072									
				4078 4080 4081 4086 4093 4093 4094 4094 4097 4104 4105 4105									
				4106 4106 4118 4129 4130 4135 4136 4141 4141* 4144* 4156 4157									
				4157 4160 4161 4161 4164 4180 4333 4333 4337 4337 4341 4345									
				4347 4351 4351 4352 4352 4353 4353 4354 4354 4355 4355 4356									
				4356 4357 4357 4365 4365 4366 4366 4367 4367 4368 4368 4369									
				4369 4370 4370 4371 4371 4372 4372 4547 4547 4551 4555 4555									
				4556 4556 4557 4557 4558 4558 4559 4559 4560 4560 4561 4561									
				4566 4570 4570 4571 4571 4572 4572 4573 4573 4574 4574 4575									
				4575 4576 4576 4577 4577 4581 4581 4585 4587 4782 4788 4791									
				4802 4803 4810 4817 4831 5005 5028 5245 5261 5266 5302 5326									
				5874* 5875 5876* 5877 5877* 5881 5954* 5955 5974 6023 6532* 6533									
				6534 6545* 6553* 6561* 6569* 6580 6613* 6614 6860* 6861 7108* 7110									
				7130* 7132 7165* 7166 7167* 7190* 7191 7201* 7202 7206* 7228* 7229									
				7233 7233* 7499 7519 7568* 7569 7575 7584 7593 7595 7602 7610									
				7618 7619 7619 7650* 7651 7656 7665 7672 7674 7678 7679 7679									
				7708* 7919* 7920 7920 7942* 7944 7946 7950 7957 7957 7983* 7984									
				7991* 8001 8004 8004 8012 8012 8013 8014* 8015* 8017 8017 8312*									
				8313 8337* 8338 8344 8370* 8371 8661* 8666 8693* 8700 8707 8718									
				8728 8730 8731 8732 8743* 8744 8759* 8760 8761* 8762 9004 9159									
				9178 9182* 9183* 9184 9191 9206 9214 9215* 9216 9219 9232* 9273									
				9327 9328* 9329 9329* 9335 9336 9336* 9338 9344 9348* 9356 9356*									
				9357 9358 9363* 9364 9364* 9365 9368 9369 9371* 9372									
@ZERO	001	0000	0062	3295 3919 5014 5311 5793 7985 8686 8744 9184 9197 9207 9222									
				9368									
@4K	001	0010	0349										
B\$ADMK	001	0001	1650										
B\$ADSW	001	159D	1649										
B\$ARMK	001	0001	1635										
B\$ARSW	001	0A45	1634										
B\$BABF	001	1D00	1440										
B\$BCKT	001	1590	1562										
B\$BDPL	001	19E8	1514										
B\$BDSA	001	19EA	1515										
B\$BINO	001	1A6A	1578										
B\$BRLN	001	19F1	1513										
B\$BROP	001	1AF7	1619										
B\$BRVA	001	19EF	1512										
B\$BRVP	001	19EE	1511										
B\$BTAB	001	1996	1510										
B\$CADR	001	1AF9	1620										
B\$CASA	001	0000	1455										
B\$CASC	001	0671	1459										
B\$CASM	001	0608	1457										
B\$CBAS	001	14BB	1585										
B\$CBFA	001	0CBC	1540										
B\$CCGT	001	0600	1465										
B\$CCLS	001	0695	1471										
B\$CCON	001	001F	1538										
B\$CDAT	001	0600	1451										
B\$CDEF	001	0600	1452										
B\$CDIM	001	0673	1453										
B\$CDUM	001	0000	1489										

CROSS REFERENCE																			
SYMBOL	LEN	VALUE	DEFN	REFERENCES													VER 15, MOD 00	06/09/20	PAGE 186
B\$CEND	001	0600	1487	1488															
B\$CEOF	001	0600	1488																
B\$CFOR	001	0600	1460																
B\$CGET	001	06A3	1468																
B\$CGSB	001	0690	1466																
B\$CGTO	001	06B3	1464																
B\$CIFA	001	0600	1462																
B\$CIFC	001	0600	1463																
B\$CIMG	001	0600	1477																
B\$CINP	001	0600	1472																
B\$CLTA	001	0000	1454																
B\$CLTC	001	0669	1458																
B\$CLTM	001	0600	1456																
B\$CMAT	001	0600	1478																
B\$CMGT	001	0665	1479																
B\$CMIN	001	06D3	1480																
B\$CMPR	001	069B	1483																
B\$CMPT	001	069B	1482																
B\$CMPU	001	0600	1484																
B\$CMRD	001	06D0	1481																
B\$CNXT	001	0600	1461																
B\$CPCT	001	0CA8	1543																
B\$CPRT	001	0600	1475																
B\$CPRU	001	0600	1476																
B\$CPSE	001	06E7	1485																
B\$CPUT	001	0600	1469																
B\$CPWA	001	0CA6	1614																
B\$CRAD	001	150D	1584																
B\$CRBS	001	1509	1586																
B\$CREA	001	06CF	1473																
B\$CREM	001	0000	1450																
B\$CRMK	001	0001	1662																
B\$CRSR	001	06E3	1474																
B\$CRST	001	06A6	1470																
B\$CRSW	001	0E42	1661																
B\$CRTN	001	06CF	1467																
B\$CSBF	001	0600	1437	1451	1452	1453	1456	1457	1458	1459	1460	1461	1462	1463	1464				
				1465	1466	1467	1468	1469	1470	1471	1472	1473	1474	1475	1476				
				1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487	1490				
				1491	1492	1493	1494												
B\$CSCN	001	14B0	1559																
B\$CSMK	001	0007	1665																
B\$CSSW	001	14BC	1664																
B\$CSTP	001	06D6	1486																
B\$CSTR	001	14CC	1583																
B\$CSXA	001	2000	1443																
B\$CTYP	001	0A5F	1537																
B\$CVPD	001	0C5D	1542																
B\$CVPG	001	0CA5	1541																
B\$CWRK	001	F500	1611																
B\$DIST	001	0700	1503																
B\$DLNK	001	1B37	1609																
B\$DL4T	001	1A6B	1580																
B\$DPWA	001	0E46	1615																

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 187

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B\$ERSW	001	0993	1637	
B\$FACA	001	0E53	1546	
B\$FAIS	001	15AC	1563	
B\$FAIW	001	15A0	1564	
B\$FCON	001	0A46	1536	
B\$FORT	001	1B0E	1605	
B\$FPWA	001	15AC	1616	
B\$FRMK	001	0007	1656	
B\$FRSW	001	16CC	1655	
B\$FSC1	001	0E4C	1547	
B\$FSC2	001	0E4D	1548	
B\$FSMK	001	0007	1647	
B\$FSSW	001	0E5C	1646	
B\$FSVA	001	0E4F	1549	
B\$FTND	001	1B0B	1607	
B\$FTPT	001	1B0D	1606	
B\$FVME	001	15A2	1568	
B\$FVMP	001	15A4	1569	
B\$FVMS	001	15A6	1570	
B\$FVPE	001	15A8	1565	
B\$FVPP	001	15AA	1566	
B\$FVPS	001	15AC	1567	
B\$GBSW	001	08AF	1640	
B\$GBWK	001	0001	1641	
B\$GETC	001	0867	1517	
B\$GPTR	001	0878	1519	
B\$GTBF	001	1E00	1441	
B\$IFMK	001	0007	1659	
B\$IFSW	001	16E5	1658	
B\$INVT	001	1B38	1599	
B\$KWMK	001	0001	1653	
B\$KWSW	001	159E	1652	
B\$LBAS	001	185E	1590	
B\$LBSV	001	18E7	1588	
B\$LDRP	001	1A00	1438	
B\$LINE	001	07D0	1505	
B\$LIST	001	1853	1572	
B\$LRTN	001	18EB	1589	
B\$LSTR	001	1862	1587	
B\$LTYP	001	18F2	1573	
B\$MATR	001	18F3	1575	
B\$MBMK	001	0007	1674	
B\$MBSW	001	1903	1673	
B\$MFBK	001	1B8F	1601	
B\$MGMK	001	0007	1671	
B\$MGSW	001	18FF	1670	
B\$MPMK	001	0007	1677	
B\$MPSW	001	1981	1676	
B\$MRMK	001	0007	1668	
B\$MRSW	001	0DDE	1667	
B\$NUMC	001	0873	1518	
B\$NXMK	001	0007	1644	
B\$NXSW	001	071D	1643	
B\$PARP	001	0A41	1526	
B\$PBNL	001	0A01	1532	
B\$PCAD	001	0A40	1527	

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 188

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B\$PCDL	001	09D3	1531	
B\$PCPG	001	0A35	1530	
B\$PECT	001	0A44	1534	
B\$PERC	001	0A39	1533	
B\$PFAE	001	0033	1524	
B\$PFCL	001	009D	1525	
B\$PFNC	001	094E	1522	
B\$PFWP	001	0015	1523	
B\$PNBY	001	0A41	1528	
B\$PPWA	001	0A35	1613	
B\$PRM1	001	1AF3	1617	
B\$PTBF	001	1F00	1442	
B\$PUTC	001	093A	1521	
B\$PVAD	001	0A43	1529	
B\$RMRK	001	1AE6	1582	
B\$RTRN	001	1AF5	1618	
B\$SABF	001	1C00	1439	
B\$SCAN	001	1514	1561	
B\$SCAT	001	13C8	1556	
B\$SCON	001	001B	1539	
B\$SCVT	001	12E0	1554	
B\$SDPL	001	07DA	1507	
B\$SFAB	001	0E48	1551	
B\$SFNT	001	143C	1557	
B\$SLDT	001	109C	1553	
B\$SLVT	001	1062	1552	
B\$SNAT	001	131A	1555	
B\$SPAT	001	07E0	1508	
B\$SSTA	001	1BAC	1603	
B\$STAS	001	061B	1492	
B\$STIF	001	0606	1494	
B\$STMA	001	061B	1493	
B\$STML	001	0600	1491	
B\$STRL	001	0600	1490	
B\$SVRB	001	0E46	1550	
B\$SYMB	001	0DBC	1545	
B\$TCD2	001	0001	1623	
B\$TLTH	001	0002	1624	1625
B\$TOD1	001	0000	1622	
B\$TOTB	001	1AF8	1625	
B\$TTAB	001	1AFA	1621	1625
B\$TYPE	001	0739	1506	
B\$WORK	001	15A0	1610	
B\$ZDBN	001	19F2	1577	
B@ABAS	001	0007	2210	7619
B@ACD1	001	0001	2207	2208 7593
B@ACD2	001	0003	2208	2209 7595 7602
B@AFLG	001	0000	2202	
B@ALLA	001	005C	2027	
B@AMAX	001	0005	2209	2210
B@BLNK	001	0040	2036	
B@BLSZ	001	0100	2161	2300 2303 2306 2321 2324
B@BREQ	001	0084	1816	
B@BRHI	001	0088	1817	
B@BRLO	001	0082	1815	
B@BRNE	001	0094	1819	

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 189

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@BRNH	001	0098	1820	
B@BRNL	001	0092	1818	
B@CADD	001	0006	1685	
B@CADF	001	0058	1726	
B@CBAS	001	0003	2213	7679
B@CBNX	001	004A	1719	
B@CBRA	001	0046	1717	
B@CBRC	001	0044	1716	
B@CBRD	001	0048	1718	
B@CBRS	001	004C	1720	
B@CCLS	001	005E	1729	
B@CCMC	001	0042	1715	
B@CCMF	001	0040	1714	
B@CCNT	001	001F	2139	
B@CCSA	001	003E	1713	
B@CDCA	001	006A	1735	
B@CDDL	001	006C	1736	
B@CDIV	001	000C	1688	
B@CDMN	001	0001	2212	2213 7665
B@CDWA	001	006E	1737	
B@CEOF	001	0070	1738	
B@CEOP	001	0068	1734	
B@CFCI	001	0016	1693	
B@CFN0	001	0012	1691	
B@CFN1	001	0014	1692	
B@CFOR	001	004E	1721	
B@CGET	001	0052	1723	
B@CHAR	001	0000	2152	
B@CHLT	001	0004	1684	
B@CIEX	001	00C5	2112	
B@CIMH	001	0066	1733	
B@CINI	001	0056	1725	
B@CIPI	001	00D7	2115	
B@CIS2	001	00E2	2118	
B@CMF1	001	0018	1694	
B@CMF2	001	001A	1695	
B@CMF3	001	001C	1696	
B@CMA	001	006B	2047	
B@CMPY	001	000A	1687	
B@CMSM	001	001E	1697	
B@CNEG	001	0010	1690	
B@CNXT	001	0050	1722	
B@COLN	001	007A	2049	
B@CPMK	001	00FF	1957	1961 1965 1966 2000
B@CPRS	001	0060	1730	
B@CPRU	001	0062	1731	
B@CPUT	001	0054	1724	
B@CPWR	001	000E	1689	
B@CRSR	001	005A	1727	
B@CRST	001	005C	1728	
B@CSA1	001	0036	1709	
B@CSA2	001	0038	1710	
B@CSB1	001	003A	1711	
B@CSC1	001	002A	1703	
B@CSD0	001	002E	1705	
B@CSD1	001	0030	1706	



## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 190

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@CSD2	001	0032	1707	
B@CSF1	001	0022	1699	
B@CSF2	001	0024	1700	
B@CSTA	001	0034	1708	
B@CSTC	001	0028	1702	
B@CSTF	001	0020	1698	
B@CSTH	001	0064	1732	
B@CSTX	001	003C	1712	
B@CSUB	001	0008	1686	
B@CSV	001	0002	1683	
B@CTYP	001	0020	2137	
B@CUSC	001	002C	1704	
B@CUSF	001	0026	1701	
B@CVAR	001	005B	2026	
B@DAMK	001	0080	2205	
B@DASA	001	00FF	1966	
B@DASC	001	0040	1970	
B@DASM	001	0038	1968	
B@DCGT	001	0050	1976	
B@DCLS	001	0054	1982	
B@DDAT	001	0024	1962	
B@DDEF	001	0034	1963	
B@DDIM	001	0004	1964	
B@DDUM	001	00FF	2000	
B@DEC0	001	00F0	2095	8728
B@DEC1	001	00F1	2096	
B@DEC2	001	00F2	2097	4104
B@DEC3	001	00F3	2098	
B@DEC4	001	00F4	2099	
B@DEC5	001	00F5	2100	
B@DEC6	001	00F6	2101	
B@DEC7	001	00F7	2102	
B@DEC8	001	00F8	2103	
B@DEC9	001	00F9	2104	
B@DEND	001	0058	1998	1999
B@DEOF	001	0058	1999	
B@DFOR	001	0028	1971	
B@DGET	001	0040	1979	
B@DGSB	001	0020	1977	
B@DGTO	001	0044	1975	
B@DIFA	001	0048	1973	
B@DIFC	001	004C	1974	
B@DIGS	001	007B	2029	
B@DIMG	001	003C	1988	
B@DINP	001	0000	1983	
B@DIVD	001	0061	2046	
B@DLTA	001	00FF	1965	
B@DLTC	001	0040	1969	
B@DLTM	001	0038	1967	
B@DL01	001	0001	2280	2283
B@DL02	001	0003	2283	2286
B@DL03	001	0005	2286	2289
B@DL04	001	0007	2289	2292
B@DL05	001	0009	2292	2295
B@DL06	001	000B	2295	2298
B@DL07	001	0045	2298	2301

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 191

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@DL08	001	0145	2301	2304
B@DL09	001	0245	2304	2307
B@DL10	001	0289	2307	2310
B@DL11	001	02C3	2310	2313
B@DL12	001	02FD	2313	2316
B@DL13	001	0337	2316	2319
B@DL14	001	0371	2319	2322
B@DL15	001	0471	2322	2325
B@DL16	001	0507	2325	
B@DMAT	001	0008	1989	
B@DMGT	001	0044	1990	
B@DMIN	001	0038	1991	
B@DMPR	001	0048	1994	
B@DMPT	001	004C	1993	
B@DMPU	001	0054	1995	
B@DMRD	001	003C	1992	
B@DNXT	001	0044	1972	
B@DPNT	001	004B	2037	
B@DPRT	001	002C	1986	
B@DPRU	001	0030	1987	
B@DPSE	001	0050	1996	
B@DPUT	001	0040	1980	
B@DREA	001	000C	1984	
B@DREM	001	00FF	1961	
B@DRSR	001	005C	1985	
B@DRST	001	0050	1981	
B@DRTN	001	005C	1978	
B@DSCY	001	0004	1953	
B@DSIF	001	001C	2002	
B@DSLT	001	0010	2001	
B@DSML	001	0010	2003	
B@DSNS	001	0018	1955	
B@DSS1	001	0000	1954	
B@DSTP	001	0054	1997	
B@DTBN	001	0010	2019	
B@DTB1	001	0050	2018	
B@DTCY	001	0009	2015	
B@DTSN	001	0010	2017	
B@DTS1	001	0040	2016	
B@DTYP	001	0040	2131	4551 5242 5245
B@DURE	001	0020	1849	
B@DVCY	001	0007	2012	
B@DVC1	001	0056	2013	
B@DWCY	001	0005	2009	
B@DWT1	001	0003	2010	
B@D1MK	001	0080	2203	
B@D2MK	001	00C0	2204	
B@EOST	001	001E	2025	
B@EQUL	001	007E	2051	
B@EXPC	001	00C5	2028	
B@FOFL	001	005C	2030	
B@FVAD	001	0001	2215	
B@GETC	001	0001	2154	
B@GETE	001	00FF	2155	
B@GETS	001	0000	2153	
B@GRTR	001	006E	2048	

CROSS REFERENCE																				
SYMBOL	LEN	VALUE	DEFN	REFERENCES												VER 15, MOD 00	06/09/20	PAGE 192		
B@ICON	001	0050	2110																	
B@LADD	001	0001	1754																	
B@LADF	001	0002	1795																	
B@LADV	001	0008	2239	2260	6883	7588														
B@LBIN	001	0002	2164	2165	2171															
B@LBNX	001	0003	1788																	
B@LBRA	001	0003	1786																	
B@LBRC	001	0004	1785																	
B@LBRD	001	0003	1787																	
B@LBRS	001	0001	1789																	
B@LCCA	001	0004	2195																	
B@LCCC	001	0001	1747	1785																
B@LCDV	001	0004	2240	2261	7660															
B@LCER	001	0001	1745	1809	5925															
B@LCFN	001	0004	2196																	
B@LCLN	001	0002	1750	1801	1802	1809	3377	3401	5974	6023										
B@LCLS	001	0001	1798																	
B@LCMC	001	0001	1784																	
B@LCMF	001	0001	1783																	
B@LCNA	001	0006	2194																	
B@LCNN	001	0001	1748	1773	1782	1794	1806	7191	7212	7984										
B@LCOP	001	0001	1744	1752	1753	1754	1755	1756	1757	1758	1759	1760	1761	1762	1763					
				1764	1765	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775					
				1776	1777	1778	1779	1780	1781	1782	1783	1784	1785	1786	1787					
				1788	1789	1790	1791	1792	1793	1794	1795	1796	1797	1798	1799					
				1800	1801	1802	1803	1804	1805	1806	1807	5875								
B@LCRV	001	0013	2238	2258																
B@LCSA	001	0002	1782																	
B@LCVA	001	0002	1746	1760	1761	1762	1763	1764	1765	1766	1767	1768	1769	1771	1772					
				1774	1775	1776	1777	1778	1779	1780	1785	1786	1787	1788	1790					
				1791	1792	1804	1805	6614	6861	7110	7115	7166	7569	7651						
B@LCXX	001	0001	1749	1781	1793	1795	1799	1800	7132	7137										
B@LDAT	001	0004	1908																	
B@LDCA	001	0003	1804																	
B@LDDL	001	0003	1805																	
B@LDDM	001	0004	2168	4788																
B@LDEF	001	0003	1909																	
B@LDIM	001	0003	1910																	
B@LDIN	001	0004	2167	2168	2169															
B@LDIV	001	0001	1757																	
B@LDMN	001	0002	2165	2194	2195	2207	2208	2209	2212	2239	2240	4804	4809	4816	4822					
				4831	4831*	4846	4847	4849	4850	4851	4853	7573	7575	7584	7593					
				7595	7602	7603	7656	7665	8001	8004	8012									
B@LDSN	001	0004	2169																	
B@LDWA	001	0002	1806	8779	8785	8786														
B@LELP	001	0010	2237																	
B@LEND	001	0003	1937																	
B@LEOF	001	0001	1807																	

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 06/09/20 PAGE 193

B@LETB	001	00C2	2064				
B@LETC	001	00C3	2065				
B@LETD	001	00C4	2066				
B@LETE	001	00C5	2067				
B@LETF	001	00C6	2068				
B@LETG	001	00C7	2069				
B@LETH	001	00C8	2070				
B@LETI	001	00C9	2071				
B@LETJ	001	00D1	2072				
B@LETK	001	00D2	2073				
B@LETL	001	00D3	2074				
B@LETM	001	00D4	2075				
B@LETN	001	00D5	2076				
B@LETO	001	00D6	2077				
B@LETP	001	00D7	2078				
B@LETQ	001	00D8	2079				
B@LETR	001	00D9	2080				
B@LETS	001	00E2	2081				
B@LETT	001	00E3	2082				
B@LETU	001	00E4	2083				
B@LETV	001	00E5	2084				
B@LETW	001	00E6	2085				
B@LETX	001	00E7	2086				
B@LETY	001	00E8	2087				
B@LETZ	001	00E9	2088				
B@LEXP	001	0008	2127				
B@LFCI	001	0003	1762				
B@LFNA	001	0002	2241	2262			
B@LFN0	001	0003	1760				
B@LFN1	001	0003	1761				
B@LFOR	001	0003	1790	8779	8785	8786	
B@LFRT	001	0004	2182	2183			
B@LGET	001	0003	1792				
B@LGSB	001	0005	1916				
B@LGTO	001	0004	1915				
B@LHLT	001	0001	1753				
B@LIEX	001	0002	2113				
B@LIFN	001	0003	2176				
B@LILP	001	0009	2235	2253	2254	2255	
B@LIMG	001	0001	1927				
B@LIMH	001	0003	1802				
B@LINI	001	0002	1794				
B@LINP	001	0005	1922				
B@LIPI	001	0003	2116				
B@LISP	001	0005	2234	2242	2248	2249	2250
B@LIS2	001	0005	2119				
B@LIVT	001	0001	2192				
B@LKCL	001	0005	1921				
B@LKFR	001	0003	1912				
B@LKGT	001	0003	1918				
B@LKIF	001	0002	1914				
B@LKON	001	0002	1947				
B@LKPT	001	0003	1919				
B@LKPU	001	000A	1926				
B@LKRR	001	0007	1924				
B@LKRT	001	0005	1920				

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 194

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@LKTO	001	0002	1941	
B@LLET	001	0003	1911	
B@LL01	001	0002	2279	2280
B@LL02	001	0002	2282	2283
B@LL03	001	0002	2285	2286
B@LL04	001	0002	2288	2289
B@LL05	001	0002	2291	2292
B@LL06	001	0002	2294	2295
B@LL07	001	003A	2297	2298
B@LL08	001	0100	2300	2301
B@LL09	001	0100	2303	2304
B@LL10	001	0044	2306	2307
B@LL11	001	003A	2309	2310
B@LL12	001	003A	2312	2313
B@LL13	001	003A	2315	2316
B@LL14	001	003A	2318	2319
B@LL15	001	0100	2321	2322
B@LL16	001	0096	2324	2325
B@LMAT	001	0003	1928	
B@LMF1	001	0003	1763	
B@LMF2	001	0003	1764	
B@LMF3	001	0003	1765	
B@LMGT	001	0006	1929	
B@LMIN	001	0008	1930	
B@LMPR	001	0008	1933	
B@LMPT	001	0006	1932	
B@LMPU	001	000D	1934	
B@LMPY	001	0001	1756	
B@LMRD	001	0007	1931	
B@LMSM	001	0003	1766	
B@LNEG	001	0001	1759	
B@LNEX	001	0004	1913	
B@LNXT	001	0003	1791	8779 8785 8786
B@LPAR	001	004D	2039	
B@LPRS	001	0002	1799	
B@LPRT	001	0005	1925	
B@LPRU	001	0002	1800	
B@LPSE	001	0005	1935	
B@LPUT	001	0002	1793	
B@LPWR	001	0001	1758	
B@LREA	001	0004	1923	
B@LREM	001	0003	1907	
B@LRSR	001	0001	1796	
B@LRST	001	0001	1797	
B@LRTN	001	0006	1917	
B@LSA1	001	0003	1778	
B@LSA2	001	0003	1779	
B@LSB1	001	0003	1780	
B@LSC1	001	0003	1772	
B@LSDF	001	0004	2162	
B@LSD0	001	0003	1774	
B@LSD1	001	0003	1775	
B@LSD2	001	0003	1776	
B@LSF1	001	0003	1768	
B@LSF2	001	0003	1769	
B@LSKW	001	0002	2178	

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 195

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@LSNO	001	0002	2171	
B@LSPT	001	0003	2186	2189
B@LSTA	001	0003	1777	
B@LSTC	001	0003	1771	
B@LSTE	001	0004	1942	
B@LSTF	001	0003	1767	
B@LSTH	001	0003	1801	
B@LSTP	001	0004	1936	
B@LSTX	001	0002	1781	
B@LSUB	001	0001	1755	
B@LSVC	001	0001	1752	
B@LTHN	001	0004	1943	
B@LTYP	001	0001	2172	
B@LUFN	001	0002	2179	
B@LUSC	001	0002	1773	
B@LUSF	001	0001	1770	
B@LVPG	001	0100	2266	2269 4997 5294
B@MINS	001	0060	2045	
B@MULT	001	005C	2042	
B@NAAR	001	001D	2230	2260 2312
B@NCAR	001	001D	2231	2261 2315
B@NCRV	001	001D	2229	2258 2309
B@NDGT	001	000A	2222	2228
B@NEQL	001	007F	2052	
B@NFRT	001	000A	2181	2183
B@NICN	001	0006	2224	2226
B@NIEL	001	0007	2226	2242 2248 2253
B@NIFN	001	0018	2175	
B@NIVR	001	0001	2225	2226
B@NIVT	001	0057	2191	
B@NLDV	001	0122	2228	2250 2255 2306
B@NLRV	001	001D	2227	2249 2254 2297
B@NLTR	001	001D	2221	2227 2228 2229 2230 2231 2232
B@NSKW	001	0004	2177	
B@NSPT	001	0028	2185	
B@NUFN	001	001D	2232	2262 2318
B@NVPG	001	0100	2265	2269
B@NXHI	001	00E3	2146	3685 3955 4196
B@NXLO	001	001E	2145	3668 3688 3907 3957 4198
B@NXZR	001	0080	2144	2145 2146 3955 3956 3957 4196 4198 4199 4788 4844
B@PLUS	001	004E	2040	
B@POWR	001	005A	2041	
B@PREC	001	0020	2133	4551 4566
B@PROD	001	0023	2242	
B@PRPL	001	0002	1829	
B@PRPN	001	0001	1828	
B@PRPR	001	0004	1831	
B@PRPS	001	0003	1830	
B@PRRC	001	0007	1834	
B@PRRL	001	0008	1835	
B@PRSL	001	0005	1832	
B@PRSS	001	0006	1833	
B@PTAB	001	0000	2187	
B@PTAD	001	0001	2188	
B@PTSA	001	0002	2189	
B@PUD1	001	0006	1845	

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 196

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@PUD2	001	0007	1846	
B@PUI0	001	0001	1839	
B@PUI1	001	0004	1840	
B@PUI2	001	0005	1841	
B@PUNL	001	0002	1843	
B@PUNS	001	0003	1844	
B@PUTM	001	0010	1848	
B@RPAR	001	005D	2043	
B@SADV	001	00E8	2260	2263
B@SAVL	001	0B76	2256	2273
B@SAVS	001	065E	2251	2272
B@SCDV	001	0074	2261	2263
B@SCLN	001	005E	2044	
B@SCRV	001	0227	2258	2272 2273
B@SDMK	001	0080	2173	
B@SEXP	001	0004	2126	
B@SFAT	001	0196	2263	2272 2273 2324
B@SFNA	001	003A	2262	2263
B@SFRT	001	0028	2183	
B@SIEL	001	003F	2253	2256
B@SIES	001	0023	2248	2251
B@SIGN	001	0010	2135	4345 4551 4587
B@SLDL	001	0A32	2255	2256
B@SLDS	001	05AA	2250	2251
B@SLVL	001	0105	2254	2256
B@SLVS	001	0091	2249	2251
B@SQUO	001	007D	2050	
B@STAT	001	0000	2125	
B@TASA	001	0012	1860	
B@TASC	001	001E	1866	
B@TASM	001	0018	1862	
B@TASS	001	007B	1867	
B@TCGT	001	0030	1875	
B@TCLS	001	0042	1881	
B@TDAT	001	0006	1856	
B@TDEF	001	0009	1857	
B@TDIM	001	000C	1858	
B@TDUM	001	0078	1899	
B@TEND	001	0072	1897	
B@TEOF	001	0075	1898	
B@TFOR	001	0021	1869	
B@TGET	001	0039	1878	
B@TGSB	001	0033	1876	
B@TGTO	001	002D	1874	
B@TIFA	001	0027	1871	
B@TIFC	001	002A	1872	
B@TIFS	001	007D	1873	
B@TIMG	001	0054	1887	
B@TINP	001	0045	1882	
B@TLTA	001	000F	1859	
B@TLTC	001	001B	1863	
B@TLTM	001	0015	1861	
B@TLTS	001	0079	1864	
B@TMAS	001	007C	1868	
B@TMAT	001	0057	1888	
B@TMGT	001	005A	1889	



## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 197

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@TMIN	001	005D	1890	
B@TMLS	001	007A	1865	
B@TMPR	001	0066	1893	
B@TMPT	001	0063	1892	
B@TMPU	001	0069	1894	
B@TMRD	001	0060	1891	
B@TNXT	001	0024	1870	
B@TPRT	001	004E	1885	
B@TPRU	001	0051	1886	
B@TPSE	001	006C	1895	
B@TPUT	001	003C	1879	
B@TRAC	001	0080	2129	4551 5259 5261 5266
B@TREA	001	0048	1883	
B@TREM	001	0003	1855	
B@TRSR	001	004B	1884	
B@TRST	001	003F	1880	
B@TRTN	001	0036	1877	
B@TSTP	001	006F	1896	
B@VMC1	001	0056	2268	
B@VMLB	001	F0CD	2273	
B@VMSB	001	F5E5	2272	
B@VMSZ	001	0000	2269	2271 2272 2273
B@VMTB	001	0000	2271	
B@ZNEG	001	00D0	2142	3575 4347 6534
B@ZPOS	001	00F0	2141	3644 3849 4096 4097 4585 4782 7944 7946 7950
CAFBC0	002	0B47	4847	4804
CAFBCR	002	0B4D	4851	4804* 4816 4822 4822*
CAFBN0	002	0B4F	4853	4809 4809* 4810* 4816* 4831
CAFBN1	001	0B41	4842	4826
CAFBN2	001	0B42	4843	4817
CAFBN3	001	0B43	4844	4791
CAFBN4	001	0AE3	4773	7709 7992 9510
CAF020	003	0AEE	4782	4774 4776
CAF030	003	0AF4	4788	
CAF040	004	0B01	4797	4783 4789
CAF050	004	0B08	4802	4792
CAF060	004	0B14	4809	
CAF070	004	0B18	4810	4802*
CAF080	004	0B1F	4816	4818
CAF090	004	0B23	4817	4803* 4826* 4827
CAF100	004	0B2A	4822	
CAF110	004	0B2E	4826	4811
CAF120	004	0B35	4831	
CAF130	004	0B39	4835	4775* 4798
CAF140	004	0B3D	4836	4777*
CPUDC0	001	0A84	4382	4341
CPUFLT	001	0A27	4327	7172 7501 8696 9506
CPUP01	001	0008	4390	4351
CPUP02	001	0009	4391	4352
CPUP03	001	0009	4392	4353
CPUP04	001	000A	4393	4354
CPUP05	001	000A	4394	4355
CPUP06	001	000B	4395	4356
CPUP07	001	000B	4396	4357
CPUP08	001	000C	4398	4365
CPUP09	001	000C	4399	4366

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 198

SYMBOL	LEN	VALUE	DEFN	REFERENCES
CPUP10	001	000D	4400	4367
CPUP11	001	000D	4401	4368
CPUP12	001	000E	4402	4369
CPUP13	001	000E	4403	4370
CPUP14	001	000F	4404	4371
CPUP15	001	000F	4405	4372
CPUSAV	001	0008	4388	4333* 4337 4345 4390 4391 4392 4393 4394 4395 4396 4398 4399 4400 4401 4402 4403 4404 4405
CPUU01	001	0001	4407	4351*
CPUU02	001	0002	4408	4352*
CPUU03	001	0003	4409	4353*
CPUU04	001	0004	4410	4354*
CPUU05	001	0005	4411	4355*
CPUU06	001	0006	4412	4356*
CPUU07	001	0007	4413	4357*
CPUU08	001	0008	4415	4365*
CPUU09	001	0009	4416	4366*
CPUU10	001	000A	4417	4367*
CPUU11	001	000B	4418	4368*
CPUU12	001	000C	4419	4369*
CPUU13	001	000D	4420	4370*
CPUU14	001	000E	4421	4371*
CPUU15	001	000F	4422	4372*
CPU010	004	0A2B	4333	
CPU020	004	0A2F	4337	
CPU030	005	0A33	4341	
CPU040	003	0A38	4345	
CPU050	004	0A41	4351	4346
CPU060	003	0A5D	4361	
CPU070	004	0A60	4365	
CPU080	004	0A80	4376	4328* 4361
CUPFLT	001	0A85	4541	7234 8763 9508
CUPP01	001	0000	4599	4555*
CUPP02	001	0001	4600	4556*
CUPP03	001	0001	4601	4557*
CUPP04	001	0002	4602	4558*
CUPP05	001	0002	4603	4559*
CUPP06	001	0003	4604	4560*
CUPP07	001	0003	4605	4561*
CUPP08	001	0004	4607	4570*
CUPP09	001	0004	4608	4571*
CUPP10	001	0005	4609	4572*
CUPP11	001	0005	4610	4573*
CUPP12	001	0006	4611	4574*
CUPP13	001	0006	4612	4575*
CUPP14	001	0007	4613	4576*
CUPP15	001	0007	4614	4577*
CUPSAV	001	0008	4597	4547* 4581
CUPU01	001	0001	4616	4555
CUPU02	001	0002	4617	4556
CUPU03	001	0003	4618	4557
CUPU04	001	0004	4619	4558
CUPU05	001	0005	4620	4559
CUPU06	001	0006	4621	4560
CUPU07	001	0007	4622	4561
CUPU08	001	0008	4624	4570

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 199

SYMBOL	LEN	VALUE	DEFN	REFERENCES
CUPU09	001	0009	4625	4571
CUPU10	001	000A	4626	4572
CUPU11	001	000B	4627	4573
CUPU12	001	000C	4628	4574
CUPU13	001	000D	4629	4575
CUPU14	001	000E	4630	4576
CUPU15	001	000F	4631	4577
CUP010	004	0A89	4547	
CUP020	003	0A8D	4551	
CUP030	004	0A90	4555	
CUP040	003	0AAC	4565	
CUP050	004	0AB2	4570	
CUP060	004	0AD2	4581	4565
CUP070	003	0AD6	4585	
CUP080	004	0ADF	4591	4542* 4586
FDIADD	004	075D	3579	3566 3580 6569 8701 9512
FDICTR	001	0818	3702	3655* 3659* 3660 3676
FDIICR	001	081A	3707	3659
FDIINC	001	0001	3696	3658
FDIINI	004	081C	3709	3588
FDIINT	001	0003	3698	3588
FDIONE	001	0001	3711	3707
FDIPL1	001	081B	3708	3574
FDISGN	001	0819	3703	3643* 3680
FDISHF	001	0001	3697	3677
FDISUB	004	0751	3570	6561 8722 9513
FDIZN1	001	0010	3712	3708
FDI010	004	076C	3588	
FDI230	004	0790	3605	3600
FDI285	004	07AA	3631	3630*
FDI299	003	07AE	3632	3599
FDI300	004	07B4	3639	3588* 3620* 3621* 3622* 3630
FDI310	003	07C6	3656	3661
FDI315	004	07E3	3676	3657
FDI320	003	07F9	3681	3589* 3604* 3611 3676*
FDI881	003	0806	3688	3686
FDI888	004	0810	3691	3579* 3669 3689
FDI890	004	0814	3692	3581*
FFIAC1	001	0001	4186	
FFIAPB	001	0001	4189	4169*
FFICNT	001	0A04	4204	4120* 4123* 4142*
FFIDIV	008	0A0E	4207	4080* 4096* 4111 4118 4136 4170
FFIDVD	004	0919	4065	6545 9517
FFIEXP	001	0002	4190	4170 4171 4174 4179 4196 4198 4199
FFIIN1	001	09FA	4195	4135 4142 4158
FFIIN2	001	09F9	4194	4129
FFIONE	001	0001	4188	
FFIPSR	002	0A03	4203	4095* 4162
FFIPTR	001	0001	4187	4141
FFIRND	001	09FD	4197	4156 4160
FFIXHI	002	09FC	4196	4171
FFIXLO	002	09FF	4198	4174
FFIXPO	001	0A05	4205	4081* 4158* 4169* 4170* 4171 4174 4179* 4180
FFIZRO	002	0A01	4199	4179
FFI001	003	0931	4078	
FFI002	004	0937	4080	4061 4066

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 200

SYMBOL	LEN	VALUE	DEFN	REFERENCES
FFI003	003	094D	4096	
FFI004	003	0976	4123	4119
FFI005	004	097F	4129	4131
FFI006	004	0983	4130	4122 4125 4143
FFI010	004	098A	4135	4137
FFI011	003	0995	4141	
FFI015	003	09A3	4155	4121* 4124*
FFI020	004	09B5	4160	4155
FFI030	003	09BD	4162	4159
FFI2DV	008	0A16	4208	4111* 4112 4112* 4130
FFI200	003	09C6	4169	4163
FFI300	004	09D7	4174	
FFI350	004	09E5	4178	4073* 4074 4172* 4173 4177*
FFI400	004	09E9	4179	4176
FFI888	004	09F1	4181	4065* 4079
FFI890	004	09F5	4182	4067*
FZICTR	001	08FE	3945	3852* 3883*
FZIEXP	001	0002	3939	3921 3922 3925 3928 3944 3954 3955 3956 3957
FZIOLOL	002	0910	3957	3925
FZIMI1	002	090A	3954	3882 3883 3900
FZIMN1	001	0001	3937	3895 3899
FZIMPY	004	082A	3831	6553 9515
FZIMS1	001	FFFF	3940	3954
FZIMUC	008	0906	3946	3838* 3849* 3870 3871 3872 3873 3874 3876 3877 3878 3879 3880
				3921
FZINZR	002	090E	3956	3928
FZIONE	001	0001	3936	3880 3919*
FZIRDR	002	0908	3950	3894
FZIRD2	001	0002	3938	3894 3894* 3950
FZISUM	002	08FD	3944	3848* 3913 3919* 3920* 3921* 3922 3925 3928* 3929
FZITAB	001	0820	3825	3832
FZIUPL	002	090C	3955	3922
FZIZRO	001	0000	3935	
FZI002	003	0848	3849	
FZI009	004	0855	3864	3884
FZI010	004	0859	3865	3864*
FZI020	003	085D	3866	3865*
FZI030	003	088B	3882	3875
FZI060	004	08A6	3899	3896
FZI065	004	08AE	3901	3898
FZI070	003	08C2	3913	3906
FZI080	003	08CB	3919	3914
FZI090	004	08E1	3925	3923
FZI100	004	08EC	3928	3926
FZI888	004	08F4	3930	3831* 3909
FZI890	004	08F8	3931	3833*
I\$ADJX	001	0D56	2564	
I\$ADST	001	0C9D	2519	
I\$BASE	001	0C60	2521	
I\$BRCN	001	117B	2573	
I\$BSET	001	119D	2572	
I\$B1SW	001	0040	2629	
I\$B2SW	001	0020	2631	
I\$CADR	001	144C	2610	
I\$CALL	001	12B1	2604	6279
I\$CBM1	001	0D43	2540	

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 201

SYMBOL	LEN	VALUE	DEFN	REFERENCES
I\$CBN1	001	0D3E	2536	
I\$CBN2	001	0D3F	2537	
I\$CBN3	001	0D40	2538	
I\$CBN4	001	0D41	2539	
I\$CFBS	001	0AE3	2587	
I\$CLFA	001	0D4A	2546	
I\$CLVA	001	0D49	2545	
I\$CL1C	001	0D46	2543	
I\$CL1F	001	0D44	2541	
I\$CL2C	001	0D47	2544	
I\$CL2F	001	0D45	2542	
I\$CPG1	001	1600	2501	3339 3339* 3432
I\$CPUF	001	0A27	2583	
I\$CSCT	001	0D5A	2559	
I\$CSSW	001	0010	2633	
I\$CSXA	001	2000	2500	
I\$CUPF	001	0A85	2585	
I\$CVAD	001	1358	2598	
I\$DATA	001	0D53	2527	
I\$DAT1	001	0D55	2528	
I\$DMSW	001	0BC1	2581	
I\$ECSW	001	0004	2637	
I\$ERRC	001	0CBC	2526	
I\$FACT	001	0DD1	2566	
I\$FADD	001	075D	2589	
I\$FATE	001	0DE6	2567	
I\$FATP	001	0DE8	2568	
I\$FDVD	001	0919	2594	
I\$FMPY	001	082A	2592	
I\$FSUB	001	0751	2590	
I\$FWRK	001	0607	2510	
I\$IMCI	001	0DCE	2557	
I\$IMLN	001	0DC6	2553	
I\$IMPT	001	0DCC	2556	
I\$INDR	001	0DC5	2552	
I\$INIT	001	0607	2509	
I\$INTR	001	0C5C	2513	
I\$IRSW	001	0CDE	2533	
I\$I700	001	0E24	2595	
I\$LBFR	001	12B6	2605	
I\$LDBR	001	1329	2602	
I\$LDXR	001	1330	2603	
I\$LOCK	001	1354	2600	
I\$MDFY	001	1349	2599	
I\$MOD4	001	130B	2596	
I\$NCPG	001	000A	2621	
I\$NDSW	001	0002	2639	
I\$NISW	001	0080	2627	
I\$NPAG	001	0C68	2514	
I\$PARM	001	0D57	2529	
I\$PGDS	001	144A	2608	
I\$PGNO	001	1449	2607	
I\$PGTB	001	14CA	2611	
I\$PLRT	001	15E2	2612	
I\$PSTK	001	15CA	2613	
I\$PUB1	001	0DC8	2554	

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 202

SYMBOL	LEN	VALUE	DEFN	REFERENCES
I\$PUB2	001	0DCA	2555	
I\$RESW	001	0CE9	2534	
I\$RNMK	001	0001	2549	
I\$RNSW	001	0D5C	2548	
I\$RTRN	001	12D3	2606	
I\$SDCT	001	0D59	2561	
I\$SDPT	001	0DD0	2558	
I\$SFCT	001	0D5A	2562	
I\$SFFO	001	0D5D	2570	
I\$SICT	001	0D5B	2563	
I\$SLLC	001	0BA1	2577	
I\$SLNG	001	0BA2	2576	
I\$SNSW	001	0001	2641	
I\$SSCT	001	0D58	2560	
I\$STAK	001	0D4E	2522	
I\$STCK	001	0B50	2575	
I\$STHA	001	0D51	2532	
I\$STKB	001	0639	2511	
I\$STKI	001	0D4F	2523	
I\$STSW	001	0008	2635	
I\$TFSW	001	0D28	2535	
I\$ULNG	001	0C3A	2580	
I\$UNLK	001	1350	2601	
I\$USTK	001	0BB0	2579	
I\$VADR	001	144A	2609	
I\$WRK1	001	0D59	2530	
I\$WRK2	001	0D5B	2531	
I\$XAD1	001	0C89	2518	
I\$XAD2	001	0C82	2517	
I\$XAD3	001	0C7B	2516	
I\$XAD4	001	0C74	2515	
I\$XERR	001	0CAB	2520	
I\$XIAR	001	0D4C	2525	
I\$XPAG	001	0C61	2524	
I@APRC	001	0006	2385	
I@APRL	001	000B	2362	
I@APRS	001	0006	2339	2385
I@ASTA	001	0000	2397	
I@ASTL	001	0020	2373	
I@ASTS	001	0000	2350	2397
I@CMEQ	001	0004	2454	7967
I@CMHI	001	0008	2455	7969
I@CMLO	001	0002	2453	7965
I@DEXP	001	0000	2432	3589 3590 3590 3604 3605 3605* 3606 3620 3621 3622 3632* 3633* 3668* 3681* 3685 3688 3900* 3907* 3920 3929* 4081 4180*
I@ICBA	001	F500	2399	
I@ICBL	001	F000	2375	
I@ICBS	001	F500	2352	2399
I@IVBA	001	F531	2400	
I@IVBL	001	F049	2376	
I@IVBS	001	F531	2353	2400
I@LCRF	001	0012	2414	2415 7920 7920 7920
I@LCRV	001	0013	2415	6058 6059 7149 7150 7207 7520 7525 7920
I@LFPZ	001	0012	2484	
I@LPFL	001	0009	2364	2367 2368
I@LPFS	001	0005	2341	2344 2345 2387

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 06/09/20 PAGE 203

I@LPFV	001	0005	2387	4333	4333	4333*	5030	5033	5328	5331	6600				
I@LPPZ	001	0003	2483												
I@LPSW	001	0080	2372												
I@LSFV	001	0008	2389												
I@LUFL	001	0010	2365	2369	3444	3959	4210								
I@LUFFS	001	0008	2342	2346	2388										
I@LUFV	001	0008	2388	2389	2424	2426	2427	2429	2430	3601	3601*	3602	3603	3603	3677*
				3679	3709	3709	3838	3851	3870	3871	3872	3873	3874	3876	3877
				3878	3879	3880	3880	3894	3946	3959	4080	4105	4106	4111	4112
				4112	4120	4123	4130	4130	4136	4156	4160	4207	4208	4210	4388
				4597	6056	6057	6061	6626	7160	7957	8666	8700	8707	8718	8762
				8778	8779	8786									
I@LXPT	001	0060	2475	3339	3339	3339*									
I@MANL	001	0001	2433	3905	4072	4078	8728								
I@MANR	001	0007	2434												
I@NCPG	001	000A	2477	3428	9074										
I@NERR	001	0000	2486	4175	5925	5983	6584	6621	6870	7713	7999	8003	8702	8723	8950
				8977	9003										
I@NXPG	001	0020	2474	2475											
I@NXPT	001	0003	2473	2475											
I@PEXL	001	0008	2368												
I@PEXP	001	0004	2392	4337	4581*										
I@PEXS	001	0004	2345	2392											
I@PMNR	001	0003	2391												
I@PMN1	001	0000	2411	4599	4600	4601	4602	4603	4604	4605	4607	4608	4609	4610	4611
				4612	4613	4614									
I@PMRL	001	0007	2367												
I@PMRS	001	0003	2344	2391											
I@PRCL	001	000F	2361	2365											
I@PRCS	001	0007	2338	2342	2384										
I@PREC	001	0007	2384	3606	3645	3645	3660	3677	3679	3851	3852	3870	3871	3872	3873
				3874	3876	3877	3878	3879	3897	3897*	3899	3899*	3900*	3908	3908
				3921	4111	4118	4136	4157	4161	4170	4341				
I@PRSW	001	0087	2396	4361	4565	7614									
I@PRTE	001	0000	2471	3359*											
I@RSE1	001	0007	2424	3601	3602*	3639*	3643	3644*	3677	3679*	3680*	3709*	3846	3847	3864
				3897*	3899*	3908	3908*	3915*	4093	4094	4097*	4118	4130*	4136*	4157*
				4161*	4164*	7957	8718	8762*							
I@RSE2	001	000F	2427	3574*	3575*	3602	3603*	3631	3645	3645*	3709	3838	3846*	3847	4080
				4093*	4094	4106*	7957	8700*							
I@RSE3	001	0017	2430	3850*	3851	3851*	3870*	3871*	3872*	3873*	3874*	3876*	3877*	3878*	3879*
				3880*	3894*	3897	3899	4104*	4105	4105*	4156*	4157	4160*	4161	8666
				8707*	8730										
I@SCOD	001	0000	2464	7132*											
I@SGNL	001	000F	2370												
I@SGNS	001	0007	2347	2394											
I@SIDX	001	0001	2465	7575	7584	7656	8001	8004	8004*	8012	8012*	8013			
I@SIGN	001	0007	2394	4347*	4585	4782	6533*	6534*	7944	7946	7950	8731	8732		
I@SPSW	001	0087	2349	2396											
I@STAT	001	0000	2408	4345	4551*	4566*	4587*	5242	5245	5259	5261*	5266*			
I@SVAD	001	0001	2463	7110*	7202	7229	7499	7519	7610*	7618*	7619*	7672*	7674*	7678*	7679*
				8017*	8313										
I@UEXP	001	0000	2410	2432	4337*	4547	4788	4791*	4802	4803					
I@UMNR	001	0007	2393	2434	4341*										
I@UMN1	001	0001	2412	2433	4407	4408	4409	4410	4411	4412	4413	4415	4416	4417	4418
				4419	4420	4421	4422	4616	4617	4618	4619	4620	4621	4622	4624



CROSS REFERENCE																					
SYMBOL	LEN	VALUE	DEFN	REFERENCES												VER 15, MOD 00		06/09/20		PAGE 204	
				4625	4626	4627	4628	4629	4630	4631											
I@UMRL	001	000F	2369	2370																	
I@UMRS	001	0007	2346	2347	2393																
I@XBRC	001	0003	2447	8338	8344																
I@XCNT	001	0001	2445	7191	7984																
I@XCOD	001	0001	2446	7132																	
I@XLNO	001	0002	2443	5974	6023																
I@XOPC	001	0000	2442	2443	2444	2445	2446	2447	5875												
I@XVAD	001	0002	2444	6614	6861	7110	7166	7569	7651	8371	8694	8760									
I@1SE1	001	0000	2423	2424	2426	3589	3590	3604	3605*	3606	3620	3621	3622	3632*	3656						
				3668*	3681*	3685	3688	3905	3907*	4078	4081	4086*	4106	4180*	7944						
I@1SE2	001	0008	2426	2427	2429	3590	3605	3633*	4072	7946	7950										
I@1SE3	001	0010	2429	2430	3895	4129*	4135*														
ICARST	001	0F84	7447																		
ICASA1	001	0F84	7454	6125																	
ICASA2	001	0F8B	7462	6126																	
ICASB1	001	0F92	7472	6127																	
ICASC1	001	0FBC	7514	6119																	
ICASF1	001	0F9D	7486	6115																	
ICASF2	001	0FA4	7494	6116																	
ICA010	004	0F96	7477	7456	7464																
ICA020	005	0FA8	7499	7488																	
ICA030	004	0FB5	7505																		
ICA040	005	0FC0	7519																		
ICA050	003	0FCD	7525																		
ICA200	001	0FD6	7553	7455	7487																
ICA210	001	0FDD	7559	7463	7495																
ICA220	004	0FE1	7564	7555																	
ICA230	003	0FE5	7568																		
ICA240	004	0FED	7573																		
ICA250	003	0FF9	7579	7554*	7560*																
ICA260	004	0FFC	7583																		
ICA270	004	1004	7588	7579																	
ICA280	004	100C	7593																		
ICA290	003	101A	7600																		
ICA300	004	101D	7602	7604																	
ICA310	004	1021	7603	7600																	
ICA320	004	1029	7610																		
ICA330	004	103C	7618	7614																	
ICA340	004	1044	7623	7564*																	
ICA400	001	1048	7645	7473	7515																
ICA410	003	104C	7650																		
ICA420	004	1054	7655																		
ICA430	004	105C	7660																		
ICA440	004	1064	7665																		
ICA450	004	106B	7672																		
ICA460	004	108B	7683	7646*																	
ICA600	001	108F	7702	7574	7583	7655															
ICA610	004	1093	7707																		
ICA620	003	109E	7713																		
ICA630	004	10A1	7714	7703*																	
ICA640	003	10A5	7718	7594	7596	7666															
ICBBNX	001	1149	8269	6135																	

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 205

SYMBOL	LEN	VALUE	DEFN	REFERENCES
ICBBRS	001	115B	8307	6136
ICBRAN	001	1149	8262	
ICBSET	001	119D	8389	8951 9495
ICB010	004	1149	8273	
ICB020	003	1150	8283	
ICB030	005	1153	8295	
ICB040	003	1158	8299	
ICB050	004	115B	8311	
ICB060	003	1166	8317	
ICB070	003	116C	8322	
ICB080	003	1172	8337	
ICB090	003	117A	8344	8338* 9496
ICB100	003	1180	8358	8273* 8359 8361 8365*
ICB110	004	1183	8365	
ICB120	003	118A	8370	8283 8299 8358
ICB130	003	1191	8375	8318
ICB140	003	1197	8380	
ICB150	004	119D	8393	8376
ICB160	004	11A4	8398	
ICB170	005	11AB	8403	8394
ICB180	004	11B4	8408	
ICELST	001	0EDB	7097	
ICESTA	001	0EDB	7104	6124
ICESTC	001	0F09	7148	6118
ICESTF	001	0F17	7159	6114
ICESTX	001	0EF2	7126	6128
ICEUSC	001	0F37	7186	6120
ICEUSF	001	0F6A	7223	6117
ICE010	003	0EDB	7108	
ICE020	003	0EE9	7115	
ICE050	003	0EF2	7130	
ICE060	003	0F00	7137	
ICE100	003	0F1E	7165	7153
ICE110	004	0F2D	7172	7151* 7161*
ICE120	003	0F31	7176	
ICE150	003	0F37	7190	
ICE160	004	0F3E	7195	
ICE170	004	0F47	7200	7213
ICE180	004	0F53	7206	7196*
ICE190	004	0F5F	7212	
ICE200	004	0F6A	7227	
ICE210	003	0F76	7233	
ICFADD	001	0E66	6568	6101
ICFAFN	001	0000	6643	6580
ICFDIV	001	0E51	6544	6104
ICFFN0	001	0E7E	6598	6107
ICFFN1	001	0E88	6607	6108
ICFLTA	001	0E32	6511	
ICFMPY	001	0E58	6552	6103
ICFNEG	001	0E3F	6529	6106
ICFPF0	001	0000	6645	6599
ICFPF1	001	0001	6646	6608
ICFPWR	001	0E32	6518	6105
ICFSCV	001	0EAC	6637	6533
ICFSUB	001	0E5F	6560	6102
ICF005	005	0E46	6533	

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 206

SYMBOL	LEN	VALUE	DEFN	REFERENCES
ICF007	003	0E4E	6536	
ICF010	004	0E6A	6579	6546 6554 6562
ICF020	003	0E71	6584	6523
ICF030	004	0E77	6589	6536
ICF100	003	0E8F	6613	6601
ICF110	002	0E9C	6617	6614*
ICF120	003	0E9D	6621	
ICF130	003	0EA3	6626	
ICLFOR	001	11C0	8653	6137
ICLFSW	001	0008	8785	8686 8744*
ICLFSZ	001	127A	8779	8768
ICLFV2	001	0017	8786	8666* 8700 8707
ICLL3F	001	1279	8778	8660
ICLNXT	001	11DC	8676	6138
ICLOOP	001	11C0	8646	
ICL010	005	11C0	8660	
ICL020	004	11CB	8666	
ICL030	004	11DC	8680	
ICL040	003	11E3	8686	
ICL050	004	11F1	8693	8687
ICL060	004	1202	8700	
ICL070	004	1212	8707	
ICL080	004	1216	8717	8669
ICL090	004	121F	8722	
ICL100	003	1229	8728	
ICL110	003	1239	8732	8730* 8731*
ICL120	003	123F	8743	
ICL130	004	124E	8750	
ICL140	003	1256	8759	8729 8733
ICL150	005	126E	8768	
ICMATF	001	0EAD	6820	
ICMLDV	001	0EDA	6883	6836 6843 6850
ICMMF1	001	0EBE	6849	6110
ICMMF2	001	0EB9	6842	6111 6829
ICMMF3	001	0EB4	6835	6112
ICMMSM	001	0EAD	6827	6113
ICM010	003	0EC3	6860	
ICM020	002	0ED0	6864	6861*
ICM030	003	0ED1	6870	
ICM040	003	0ED7	6875	
ICTCMC	001	10AB	7914	6131
ICTCMF	001	10BF	7935	6130
ICTCSA	001	1102	7978	6129
ICTEST	001	10AB	7907	
ICT010	004	10AB	7918	
ICT020	003	10B6	7925	
ICT030	004	10BF	7941	
ICT032	003	10C6	7944	
ICT035	003	10D9	7950	7945
ICT040	004	10E3	7957	7949
ICT045	003	10EA	7959	7948* 7952*
ICT050	004	10ED	7965	7925 7947
ICT060	004	10F4	7967	7926 7958
ICT070	004	10FB	7969	7927 7951 7959
ICT080	003	1102	7983	
ICT090	004	110C	7990	

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 207

SYMBOL	LEN	VALUE	DEFN	REFERENCES
ICT100	003	1117	7999	
ICT110	003	1124	8003	8000
ICT120	004	112B	8010	8002
ICT130	004	1142	8017	8013*
ICVADN	001	0000	9016	9004
ICVFCI	001	127B	8939	6109
ICVFIO	001	129A	8990	6139 6140 6141 6142 6143 6144 6145 6146 6147
ICVMEX	001	127B	8932	
ICVSDN	001	128B	8963	6121 6122 6123
ICV010	004	127B	8943	
ICV020	003	1281	8950	
ICV030	003	1288	8955	
ICV040	004	128B	8970	
ICV050	003	1291	8977	
ICV060	003	1297	8982	
ICV070	004	129A	8994	
ICV080	003	12A8	9003	
ICV090	003	12AE	9008	
IMIBN1	001	06CE	3400	3307 3308 3319 3361
IMIEX1	001	06D2	3408	3318* 3319* 3320 3321 3323 3331 3333 3335 3336 3337
IMIEX2	001	06D4	3411	3320* 3321* 3322 3334
IMIEX3	001	06D6	3414	3322* 3323* 3329 3330
IMIHLN	002	06D0	3401	3377
IMINIT	001	0607	3289	3441 9432
IMIPAD	002	06DC	3430	3308*
IMIPCT	001	06DA	3426	3306* 3307* 3357
IMIPDC	001	06D8	3423	
IMIPDF	001	06D7	3422	
IMIPDP	001	06D7	3420	3314 3352
IMIPDS	001	06D9	3424	
IMISTB	001	0639	3446	9434
IMIWRK	001	0607	3443	9433
IMI010	004	060B	3295	3290 3291
IMI020	005	0612	3306	
IMI030	004	061F	3313	
IMI040	005	0625	3318	
IMI050	004	0681	3351	3296
IMI060	004	0687	3356	3341
IMI070	003	068F	3359	3357* 3361* 3362
IMI080	004	069C	3366	
IMI090	004	06A3	3371	
IMI100	006	06A7	3375	3367
IMI120	004	06BE	3386	
IMI130	004	06CA	3392	
IMM110	004	06B8	3381	
INTADS	001	0C9D	5889	6627 7116 7138 7176 7526 9442
INTAD1	001	0C89	5858	6230 6590 7237 7966 7968 7970 9441
INTAD2	001	0C82	5845	7140 7215 8019 9440
INTAD3	001	0C7B	5832	5942 6027 6035 6042 6629 6871 7118 7178 7478 7506 7527 8366
				8978 9439
INTAD4	001	0C74	5819	8345 9438
INTBAT	001	0D5D	6097	5876 9493
INTBM1	002	0D43	6054	9463
INTBN1	002	0D3E	6050	5862 6247 9459
INTBN2	001	0D3F	6051	5849 9460
INTBN3	001	0D40	6052	5836 9461

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 208

SYMBOL	LEN	VALUE	DEFN	REFERENCES
INTBN4	001	0D41	6053	5823 9462
INTDAT	002	0D53	6077	9450
INTDT1	002	0D55	6078	9451
INTEOP	001	0DFB	6238	6150
INTERC	004	0CBC	6200	5900* 5962* 5973* 5983* 9449
INTERP	001	0C5C	5782	3392 9436
INTERR	001	0CAB	5908	5251 5964 5976 6585 6622 6875 7719 8323 8381 8689 8703 8724 8955 8982 9008 9443
INTFAP	002	0DE8	6187	9491
INTFAT	001	0DD1	6182	6189 9489
INTFTE	001	0DE6	6185	9490
INTHLT	001	0DE9	6218	6100
INTIAR	002	0D4C	6067	5810* 5823* 5836* 5849* 5862* 5874 5954 9448
INTIMH	001	0CC7	5937	6149
INTISW	003	0CDE	6202	9456
INTLFA	001	0D4A	6061	9469
INTLVA	002	0D49	6060	9468
INTL1C	001	0D46	6058	9466
INTL1F	001	0D44	6056	9464
INTL2C	001	0D47	6059	9467
INTL2F	001	0D45	6057	9465
INTNFA	001	000A	6180	6184
INTPAG	001	0C68	5802	8410 9437
INTPAR	002	0D57	6080	6172 9452
INTPB1	002	0DC8	6160	9477
INTPB2	002	0DCA	6161	9478
INTPCC	002	0D5A	6166	6167 6168 6169 6170 9482
INTPC1	002	0DCE	6163	9480
INTPDC	002	0D59	6168	9484
INTPDP	002	0DD0	6164	9481
INTPFC	002	0D5A	6169	9485
INTPIC	002	0D5B	6170	9486
INTPIL	001	0DC6	6159	9476
INTPIN	001	0DC5	6158	9475
INTPIP	002	0DCC	6162	9479
INTPJX	002	0D56	6172	9487
INTPSC	002	0D58	6167	9483
INTRND	001	0D5C	6085	6086 9471
INTRNM	001	0001	6088	9472
INTRSW	003	0CE9	6205	9457
INTSHA	002	0D51	6075	5955* 5956* 9455
INTSTH	001	0CD0	5950	6148
INTSTI	001	0D4F	6073	5894 9446
INTSTP	002	0D4E	6069	5894* 9445
INTSVC	001	0E0C	6256	6015 6099
INTTSW	003	0D2B	6208	9458
INTWK1	002	0D59	6082	9453
INTWK2	002	0D5B	6083	6166 9454
INTXAD	001	0000	6197	5881
INTXEC	001	0C8D	5870	5811 5824 5837 5850 8399 8770
INTXPG	004	0C61	6199	5956 6242 6247* 9447
INT010	004	0C60	5788	5783 5784 5789 5791 6199 6248 9444
INT020	004	0C68	5806	
INT030	005	0C6C	5810	
INT040	004	0C74	5823	
INT050	004	0C7B	5836	

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 209

SYMBOL	LEN	VALUE	DEFN	REFERENCES
INT060	004	0C82	5849	
INT070	004	0C89	5862	
INT080	003	0C8D	5874	
INT090	003	0C97	5877	5875*
INT100	003	0C9A	5881	
INT110	004	0CA0	5894	
INT120	004	0CA4	5896	5890*
INT130	003	0CA8	5900	
INT140	004	0CAB	5912	
INT150	004	0CBB	5923	5924 5926 6200
INT200	003	0CD0	5954	
INT210	003	0CDD	5964	5939 5940* 5965 5967 6202
INT220	003	0CE8	5976	5977 5979 5982* 6205
INT230	003	0CF1	5987	5988 5990 6010* 6228*
INT240	004	0CF4	5994	
INT250	004	0D03	5999	
INT260	004	0D0A	6004	
INT270	003	0D16	6010	5987
INT280	004	0D19	6014	5997 6000
INT285	004	0D21	6019	
INT290	005	0D25	6023	
INT300	003	0D2A	6027	6028 6030 6208
INT310	004	0D2D	6034	
INT320	004	0D34	6039	
INT400	004	0DED	6223	
INT410	003	0DF5	6228	
INT500	005	0DFB	6242	
INT510	004	0E04	6247	
INT600	004	0E10	6261	
INT610	004	0E1C	6271	
INT700	004	0E24	6278	5912 6005 6219 6257
INT710	004	0E2E	6281	6278*
IPGAIT	002	1444	9306	9282
IPGBAS	005	13A8	9313	9173 9176 9221
IPGBFR	002	12D2	9094	9088
IPGB01	001	1452	9309	9349
IPGB02	002	130A	9112	9122
IPGB04	002	1308	9111	9099 9103 9123
IPGB24	001	1443	9291	9259 9262
IPGB32	002	1442	9290	9258 9261
IPGCAD	002	144C	9301	9192* 9201 9236 9530
IPGCAL	001	130B	9116	5351 5913 5918 6039 6261 6266 6521 6616 6863 8943 8970 8995
				9093 9099* 9100 9118 9120 9123*
IPGCA2	002	1454	9310	9272* 9273*
IPGCPG	001	144B	9300	9190* 9191* 9198
IPGCVA	001	1358	9174	4993 5022 9146 9152 9519
IPGDAD	002	1451	9308	9251* 9253* 9254 9254* 9255 9255* 9256* 9257* 9258* 9261* 9264 9266*
				9267* 9330* 9338 9344* 9369* 9370* 9379
IPGDPL	001	144F	9307	9242* 9278 9378*
IPGDXR	002	1445	9292	9306 9348 9411
IPGHE0	002	1447	9293	9267
IPGKAL	001	12B1	9081	9525
IPGLBR	001	1329	9131	9523
IPGLBT	001	0001	9075	9108 9124 9167 9335 9358
IPGLOK	001	1354	9166	5806 9521
IPGLRT	001	15E2	9394	9107 9215 9328 9399 9532



CROSS REFERENCE																		
SYMBOL	LEN	VALUE	DEFN	REFERENCES												VER 15, MOD 00	06/09/20	PAGE 210
IPGLUV	001	0002	9073	9210	9226	9228	9228*	9302	9303	9304	9330	9338	9344	9400	9402			
IPGLXR	001	1330	9137	9524														
IPGMOD	001	1349	9150	5226	5315	8668	8746	9520										
IPGMOV	001	13E8	9250	9252	9380													
IPGMX1	002	1445	9411	3333*														
IPGMX2	006	13C4	9412	3334*														
IPGMX3	003	145D	9413	3335*														
IPGMX4	003	146A	9414	3336*														
IPGMX5	003	1464	9415	3337*														
IPGNMX	001	000A	9074	9228	9228	9228*	9292	9329	9331	9336	9395	9399	9400	9402				
IPGRBT	001	0002	9076	9151	9357	9358												
IPGRED	001	13DE	9241	9280	9373													
IPGRTN	001	12D3	9098	9526														
IPGSLT	001	1455	9326	9185														
IPGSTK	001	15CA	9389	9119	9533													
IPGSZ1	004	12EA	9404	3325*														
IPGSZ2	003	1373	9405	3326*														
IPGSZ3	003	1423	9406	3327*														
IPGTBL	001	14CA	9384	9182	9363	9531												
IPGULK	001	1350	9156	6243	8404	9089	9158	9160	9522									
IPGUTA	002	143E	9288	9209	9409													
IPGUT1	006	13C4	9408	3329*	3330*													
IPGUT2	002	143E	9409	3331*														
IPGUVL	002	144E	9302	9210	9226*	9330												
IPGUVT	001	15EC	9399	9228	9228*	9288	9401											
IPGVAD	002	144A	9299	9088*	9145*	9192	9528	9529										
IPGVMS	002	1440	9289	9256														
IPGVPG	001	1449	9298	9183	9243	9527												
IPGVRT	001	13E9	9252	9243*	9259*	9262*	9379*											
IPG100	004	12EA	9104	9404														
IPG120	003	12FC	9108	9104*	9105*													
IPG140	004	12FF	9109	9106*														
IPG160	004	1303	9110	9101*														
IPG170	003	12B5	9083	9084	9086	9090*												
IPG175	004	12CA	9092	9083	9087													
IPG180	004	1334	9142	9133														
IPG200	006	1340	9145	9082*	9092	9143*												
IPG220	004	1366	9182	9283	9370													
IPG240	003	1373	9190	9405														
IPG260	003	137E	9196	9132*	9199*													
IPG280	003	138C																



## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 211

SYMBOL	LEN	VALUE	DEFN	REFERENCES
IPG600	003	1464	9331	9415
IPG620	003	1467	9335	9350
IPG640	003	146A	9336	9414
IPG660	004	1470	9338	9331* 9343 9349*
IPG680	004	147B	9344	9343* 9355 9362 9372
IPG700	003	147F	9348	9337 9339
IPG720	003	148D	9356	9355*
IPG740	003	149E	9364	9366
IPG760	003	14A1	9365	9362*
IPG780	004	14B2	9371	9327*
ISTACK	001	0B50	4968	7168 7500 7521 7589 7661 8695 9498
ISTBN1	001	0BAF	5047	5013
ISTLLC	004	0BA1	5055	9500
ISTLNG	004	0BA2	5053	9499
IST010	004	0B5B	4977	4969 4971 5006
IST020	005	0B5F	4982	
IST030	004	0B6B	4993	
IST040	003	0B6F	4997	
IST050	004	0B7B	5004	
IST060	004	0B7F	5005	4997* 4998* 4999 4999*
IST070	003	0B87	5010	4982* 4983*
IST080	004	0B97	5022	4985
IST090	004	0B9B	5026	
IST100	004	0B9F	5028	4977 4977* 4983 5010* 5011 5011* 5029 5031 5033* 5053 5055
IST110	004	0BA3	5033	
IST120	004	0BA7	5038	4970*
IST130	004	0BAB	5039	4972*
IUSBN1	001	0C5B	5362	5310
IUSDSW	003	0BC1	5371	9504
IUSLNG	004	0C3A	5368	9503
IUSTAK	001	0BB0	5219	7208 7235 8764 9502
IUS010	004	0BB8	5226	5220 5260 5303
IUS012	004	0BBC	5232	
IUS014	003	0BC0	5234	5235 5237 5371
IUS016	004	0BC3	5241	
IUS018	003	0BD1	5245	5243
IUS020	003	0BD4	5246	5241* 5244*
IUS022	004	0BD7	5250	
IUS025	003	0BDE	5259	5234 5246
IUS030	003	0BEB	5266	
IUS040	004	0BF7	5277	5262
IUS050	005	0BFB	5282	
IUS060	003	0C07	5294	
IUS070	004	0C13	5301	
IUS080	004	0C17	5302	5294* 5295* 5296 5296*
IUS090	003	0C1F	5307	5282* 5283*
IUS100	004	0C2F	5315	
IUS110	004	0C33	5324	5285
IUS120	004	0C37	5326	5277 5277* 5283 5307* 5308 5308* 5327 5329 5331* 5368
IUS130	004	0C3B	5331	
IUS140	004	0C3F	5336	5221*
IUS150	003	0C43	5338	5268* 5339 5341 5345*
IUS160	004	0C46	5345	
IUS170	004	0C51	5351	
IUS175	004	0C57	5354	5222* 5338 5347
IZADST	001	0C9D	9442	

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 212

SYMBOL	LEN	VALUE	DEFN	REFERENCES
IZADUX	002	0D56	9487	
IZBASE	004	0C60	9444	6512 6821 7098 7111 7133 7448 7908 8263 8647 8717 8933
IZBRCN	003	117B	9496	7965* 7967* 7969*
IZBSET	001	119D	9495	
IZCADR	002	144C	9530	5004 5026 5232 5301 5324 5810
IZCALL	001	12B1	9525	
IZCBM1	002	0D43	9463	8015
IZCBN1	002	0D3E	9459	7212 7603
IZCBN2	001	0D3F	9460	
IZCBN3	001	0D40	9461	8680 8750
IZCBN4	001	0D41	9462	
IZCFBS	001	0AE3	9510	
IZCLFA	001	0D4A	9469	7227
IZCLVA	002	0D49	9468	7200 7477 8295 8311
IZCL1C	001	0D46	9466	7195
IZCL1F	001	0D44	9464	6530 6589 6609 6828 7505 7707 7990
IZCL2C	001	0D47	9467	7918
IZCL2F	001	0D45	9465	6519 6579 7941
IZCPUF	001	0A27	9506	
IZCSCT	002	0D5A	9482	
IZCUPF	001	0A85	9508	
IZCVAD	001	1358	9519	
IZDATA	002	0D53	9450	3375*
IZDATI	002	0D55	9451	3376*
IZDMSW	003	0BC1	9504	8994* 8997*
IZERRC	004	0CBC	9449	3687* 3690* 3924* 3927* 4175* 4178* 4797* 5250* 6584 6621 6870 7713 7718* 7999 8003* 8322* 8380* 8688* 8702 8723 8950 8977 9003
IZFACT	001	0DD1	9489	
IZFADD	004	075D	9512	
IZFATE	001	0DE6	9490	
IZFATP	002	0DE8	9491	8295*
IZFDVD	004	0919	9517	
IZFMPY	004	082A	9515	
IZFSUB	004	0751	9513	
IZFWRK	001	0607	9433	3601* 3603 3677* 3679
IZIMC1	002	0DCE	9480	
IZIMLN	001	0DC6	9476	
IZIMPT	002	0DCC	9479	
IZINDR	001	0DC5	9475	
IZINIT	001	0607	9432	
IZINTR	001	0C5C	9436	
IZIRSW	003	0CDE	9456	8279*
IZLDBR	001	1329	9523	
IZLDXR	001	1330	9524	
IZLOCK	001	1354	9521	
IZMDFY	001	1349	9520	
IZNPAG	001	0C68	9437	
IZPARM	002	0D57	9452	5267* 6599* 6608*
IZPGDS	002	144A	9528	4982 4998 5014* 5282 5295 5311* 5793*
IZPGNO	001	1449	9527	5013* 5310* 5788* 6242* 8403* 8667* 8745*
IZPGTB	001	14CA	9531	3356
IZPLRT	001	15E2	9532	
IZPSTK	001	15CA	9533	
IZPUB1	002	0DC8	9477	
IZPUB2	002	0DCA	9478	
IZRESW	003	0CE9	9457	8769*

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 06/09/20 PAGE 213

IZRNMK	001	0001	9472	
IZRNSW	001	0D5C	9471	
IZRTRN	001	12D3	9526	
IZSDCT	002	0D59	9484	
IZSDPT	002	0DD0	9481	
IZSFCT	002	0D5A	9485	
IZSFFO	001	0D5D	9493	
IZSICT	002	0D5B	9486	
IZSLLC	004	0BA1	9500	6600*
IZSLNG	004	0BA2	9499	7149* 7520* 7588* 7660*
IZSSCT	002	0D58	9483	
IZSTAK	002	0D4E	9445	3570 3582 3667 3678 3834 3901 4068 4144 6519* 6530* 6532 6579* 6589* 6609* 6828* 6836* 6843* 6850* 7109 7131 7167 7195* 7196 7200* 7201 7227* 7228 7477* 7505* 7707* 7708 7918* 7919 7941* 7942 7990* 7991 8010* 8011* 8014 8311* 8312 8317 8660* 8661 8693 8761
IZSTCK	001	0B50	9498	
IZSTHA	002	0D51	9455	
IZSTKB	001	0639	9434	6071 8317 8718* 8762
IZSTKI	001	0D4F	9446	6626* 7115* 7137* 7150* 7160* 7525*
IZTFSW	003	0D2B	9458	3371*
IZULNG	004	0C3A	9503	7207*
IZUNLK	001	1350	9522	
IZUSTK	001	0BB0	9502	
IZVADR	002	144A	9529	5267 7166* 7202* 7229* 7499* 7519* 7569* 7651* 8409* 8694* 8760*
IZWRK1	002	0D59	9453	7191* 7212* 7573 7573* 7584* 7593 7603* 7656* 7665 7672 7673 7673* 7674 7675 7675* 7676 7676* 7677 7677* 7678 7984* 7985* 8001 8010 8011 8313* 8371* 8375 8393 8398 8408 8409
IZWRK2	002	0D5B	9454	7575* 7595 7602* 7610 7611 7611* 7612 7612* 7616 7616* 7618
IZXAD1	001	0C89	9441	
IZXAD2	001	0C82	9440	
IZXAD3	001	0C7B	9439	
IZXAD4	001	0C74	9438	
IZXERR	001	0CAB	9443	
IZXIAR	002	0D4C	9448	6613 6860 7108 7130 7165 7190 7568 7650 7983 8337 8370 8398* 8662 8680* 8681 8743 8750* 8759 8768*
IZXPAG	004	0C61	9447	8393 8403 8408* 8667 8745
V\$APWR	001	0800	2692	2838 6522
V\$BFR1	001	5400	2755	2946
V\$BFR2	001	5500	2756	2947
V\$CBNZ	001	0CB2	2764	2845
V\$CCON	001	5120	2771	2943
V\$CDCV	001	3100	2768	2898
V\$CDSY	001	2E00	2767	2895
V\$CFPZ	001	0C70	2762	2844
V\$CNXZ	001	0470	2765	2833
V\$CSSR	001	5100	2770	2942
V\$CZFP	001	04AD	2763	2834
V\$DTLN	001	4600	2777	2930 6040
V\$DTVR	001	4700	2778	2931 5352
V\$FABS	001	1761	2663	2862
V\$FACS	001	1400	2679	2854
V\$FASN	001	1413	2678	2855
V\$FATN	001	1100	2677	2851
V\$FCOS	001	0A00	2674	2840
V\$FCOT	001	0D00	2672	2846
V\$FCSC	001	1725	2676	2861

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 214

SYMBOL	LEN	VALUE	DEFN	REFERENCES
V\$FDEG	001	17DA	2683	2866
V\$FDET	001	4540	2686	2929
V\$FEXP	001	0500	2670	2835
V\$FHCS	001	1500	2682	2856
V\$FHSN	001	1557	2681	2857
V\$FHTN	001	1593	2680	2858
V\$FINT	001	176C	2664	2863
V\$FLGT	001	0200	2668	2828
V\$FLOG	001	0219	2667	2830
V\$FLTW	001	020B	2669	2829
V\$FRAD	001	17CB	2684	2865
V\$FRND	001	1800	2685	2867
V\$FSEC	001	1700	2675	2860
V\$FSGN	001	17A7	2665	2864
V\$FSIN	001	0A1A	2673	2841
V\$FSQR	001	0900	2666	2839
V\$FTAN	001	0D28	2671	2847
V\$IFCI	001	1B00	2655	2871 8944
V\$IFIO	001	1A00	2657	2870 8996
V\$ISDN	001	1900	2656	2868 8971
V\$KBTL	001	1EAC	2799	
V\$KBTS	001	0DAC	2798	
V\$LPRB	001	4F00	2753	2940
V\$LPRT	001	4D00	2751	2938
V\$LPR2	001	4E00	2752	2939
V\$MADD	001	4007	2700	2918
V\$MASN	001	43A0	2698	2925
V\$MCON	001	4324	2705	2923
V\$MIDN	001	4300	2706	2922
V\$MINV	001	4500	2710	2928
V\$MMPY	001	4100	2702	2919
V\$MSMY	001	4264	2703	2921
V\$MSUB	001	4000	2701	2917
V\$MTRN	001	4400	2709	2927
V\$MZER	001	432B	2707	2924
V\$PCH1	001	5200	2791	2944
V\$PCH2	001	5300	2792	2945
V\$SCDI	001	2A00	2748	2889
V\$SCDO	001	2A96	2749	2890
V\$SFA2	001	5000	2733	2941
V\$SFD1	001	0000	2743	2826
V\$SFD2	001	0100	2744	2827
V\$SKEY	001	2500	2747	2884
V\$SPRT	001	2800	2746	2887
V\$VMPL	001	4C06	2785	2937
V\$VMPS	001	4C00	2784	2936 5919 6267
V\$XKAF	001	1C00	2732	2872
V\$XKCA	001	2400	2736	2880 5914 6262
V\$XKCL	001	240A	2735	2881
V\$XKIN	001	2B00	2731	2891
V\$XKLP	001	24AD	2737	6280
V\$XKRS	001	240D	2734	2882
V\$XMGT	001	3E06	2725	2912
V\$XMIN	001	3D00	2724	2910
V\$XMPL	001	3F06	2728	2915
V\$XMPS	001	3F00	2727	2914

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 215

SYMBOL	LEN	VALUE	DEFN	REFERENCES
V\$XMPT	001	3E0C	2726	2913
V\$XMPU	001	3F13	2729	2916
V\$XMRD	001	3E00	2723	2911
V\$XSGT	001	2100	2718	2877
V\$XSIN	001	2B6E	2717	2892
V\$XSPR	001	3400	2720	2901
V\$XSPT	001	1D00	2719	2873
V\$XSPU	001	3800	2721	2905
V\$XSRD	001	3300	2716	2900
V\$00E1	001	0000	2826	
V\$01E1	001	0100	2827	
V\$02E1	001	0200	2828	
V\$02E2	001	020B	2829	
V\$02F3	001	0219	2830	
V\$03CC	001	0300	2831	
V\$04CC	001	0400	2832	
V\$04E1	001	0470	2833	
V\$04E2	001	04AD	2834	
V\$05E1	001	0500	2835	
V\$06CC	001	0600	2836	
V\$07CC	001	0700	2837	
V\$08E1	001	0800	2838	
V\$09E1	001	0900	2839	
V\$10E1	001	0A00	2840	
V\$10E2	001	0A1A	2841	
V\$11CC	001	0B00	2842	
V\$12CC	001	0C00	2843	
V\$12E1	001	0C70	2844	
V\$12E2	001	0CB2	2845	
V\$13E1	001	0D00	2846	
V\$13E2	001	0D28	2847	
V\$14CC	001	0E00	2848	
V\$15CC	001	0F00	2849	
V\$16CC	001	1000	2850	
V\$17E1	001	1100	2851	
V\$18CC	001	1200	2852	
V\$19CC	001	1300	2853	
V\$20E1	001	1400	2854	
V\$20E2	001	1413	2855	
V\$21E1	001	1500	2856	
V\$21E2	001	1557	2857	
V\$21E3	001	1593	2858	
V\$22CC	001	1600	2859	
V\$23E1	001	1700	2860	
V\$23E2	001	1725	2861	
V\$23E3	001	1761	2862	
V\$23E4	001	176C	2863	
V\$23E5	001	17A7	2864	
V\$23E6	001	17CB	2865	
V\$23E7	001	17DA	2866	
V\$24E1	001	1800	2867	
V\$25E1	001	1900	2868	
V\$26E1	001	1A00	2870	
V\$27E1	001	1B00	2871	
V\$28E1	001	1C00	2872	
V\$29E1	001	1D00	2873	

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 216

SYMBOL	LEN	VALUE	DEFN	REFERENCES
V\$30CC	001	1E00	2874	
V\$31CC	001	1F00	2875	
V\$32CC	001	2000	2876	
V\$33E1	001	2100	2877	
V\$34CC	001	2200	2878	
V\$35CC	001	2300	2879	
V\$36CC	001	2400	2883	
V\$36E1	001	2400	2880	
V\$36E2	001	240A	2881	
V\$36E3	001	240D	2882	
V\$37E1	001	2500	2884	
V\$38CC	001	2600	2885	
V\$39CC	001	2700	2886	
V\$40E1	001	2800	2887	
V\$41CC	001	2900	2888	
V\$42E1	001	2A00	2889	
V\$42E2	001	2A96	2890	
V\$43E1	001	2B00	2891	
V\$43E2	001	2B6E	2892	
V\$44CC	001	2C00	2893	
V\$45CC	001	2D00	2894	
V\$46E1	001	2E00	2895	
V\$47CC	001	2F00	2896	
V\$48CC	001	3000	2897	
V\$49E1	001	3100	2898	
V\$50CC	001	3200	2899	
V\$51E1	001	3300	2900	
V\$52E1	001	3400	2901	
V\$53CC	001	3500	2902	
V\$54CC	001	3600	2903	
V\$55CC	001	3700	2904	
V\$56E1	001	3800	2905	
V\$57CC	001	3900	2906	
V\$58CC	001	3A00	2907	
V\$59CC	001	3B00	2908	
V\$60CC	001	3C00	2909	
V\$61E1	001	3D00	2910	
V\$62E1	001	3E00	2911	
V\$62E2	001	3E06	2912	
V\$62E3	001	3E0C	2913	
V\$63E1	001	3F00	2914	
V\$63E2	001	3F06	2915	
V\$63E3	001	3F13	2916	
V\$64E1	001	4000	2917	
V\$64E2	001	4007	2918	
V\$65E1	001	4100	2919	
V\$66CC	001	4200	2920	
V\$66E1	001	4264	2921	
V\$67E1	001	4300	2922	
V\$67E2	001	4324	2923	
V\$67E3	001	432B	2924	
V\$67E4	001	43A0	2925	
V\$68E1	001	4400	2927	
V\$69E1	001	4500	2928	
V\$69E2	001	4540	2929	
V\$70E1	001	4600	2930	

## CROSS REFERENCE

VER 15, MOD 00 06/09/20 PAGE 217

SYMBOL	LEN	VALUE	DEFN	REFERENCES
V\$71E1	001	4700	2931	
V\$72CC	001	4800	2932	
V\$73CC	001	4900	2933	
V\$74CC	001	4A00	2934	
V\$75CC	001	4B00	2935	
V\$76E1	001	4C00	2936	
V\$76E2	001	4C06	2937	
V\$77CC	001	4D00	2938	
V\$78CC	001	4E00	2939	
V\$79CC	001	4F00	2940	
V\$80E1	001	5000	2941	
V\$81E2	001	5100	2942	
V\$81E3	001	5120	2943	
V\$82E1	001	5200	2944	
V\$83E2	001	5300	2945	
V\$84E1	001	5400	2946	
V\$85E2	001	5500	2947	
V@CDPT	001	0007	2958	
V@CHGH	001	0008	3063	
V@CMIC	001	0002	2959	
V@CMNI	001	00FF	2956	
V@CMUL	001	0007	3064	
V@CNIX	001	0080	2957	
V@COEX	001	001E	2954	
V@CPLS	001	00F0	2961	
V@CPRC	001	000A	2963	
V@CSQR	001	0003	3061	
V@CSTR	001	0002	3062	
V@CTTA	001	0027	2964	
V@DCAD	001	0002	2984	2985
V@DEXP	001	0000	2989	
V@DMAN	001	000D	2991	2992
V@DMN1	001	0001	2990	
V@DPDF	001	0002	2979	
V@DSAD	001	0001	2980	
V@DSGN	001	000D	2992	
V@DVAD	001	0004	2985	
V@EART	001	0001	2962	
V@ECRT	001	0038	3035	
V@EFUL	001	00F8	3034	
V@EINV	001	00FB	3030	
V@EIPR	001	00F5	3031	
V@ENSV	001	00F7	3032	
V@ENUL	001	0000	3029	
V@ERPC	001	0020	2960	
V@ESAV	001	00F6	3033	
V@FEHN	001	0002	3059	
V@FEPL	001	0091	3055	
V@FERS	001	0003	3058	
V@FPGS	001	0081	3054	
V@FRET	001	0015	3057	
V@FSPC	001	0040	3056	
V@FTAB	001	0000	3060	
V@KADD	001	004E	3045	
V@KCLE	001	006E	3042	
V@KDIV	001	0061	3048	



## CROSS REFERENCE

SYMBOL   LEN   VALUE   DEFN   REFERENCES   VER 15, MOD 00   06/09/20   PAGE 218

V@KEMN	001	006C	3040	
V@KEPL	001	006B	3039	
V@KMUL	001	005C	3047	
V@KPER	001	004B	3050	
V@KPST	001	007B	3044	
V@KPWR	001	005A	3049	
V@KSQR	001	006F	3041	
V@KSTO	001	006D	3043	
V@KSUB	001	0060	3046	
V@LAIP	001	0003	3010	3011
V@LDEX	001	0002	3013	
V@LETE	001	0003	3017	
V@LEXP	001	0001	3007	3009
V@LFKO	001	0006	3012	
V@LINI	001	0200	3016	
V@LLKS	001	0010	3009	
V@LMAN	001	000F	3008	3009
V@LNOP	001	0015	3014	
V@LTBE	001	0007	3011	
V@LVPG	001	0100	3015	3016
V@MCHS	001	00C0	2996	
V@MCRD	001	0010	2972	
V@MDEF	001	0008	2973	
V@MEXC	001	0080	2970	
V@MEXT	001	0004	2999	
V@MICC	001	0010	2955	
V@MIPC	001	0080	2997	
V@MIPL	001	0020	3003	
V@MLST	001	0040	2971	
V@MPND	001	0000	3002	
V@MPOF	001	0080	3000	
V@MPRC	001	0020	2969	
V@MSFU	001	0002	2974	
V@MSTN	001	0004	2968	
V@OALL	001	00F4	3025	
V@ONUL	001	00F0	3021	3022
V@OPM1	001	00F2	3023	3024
V@ORTN	001	00F1	3022	3023
V@OSTK	001	00F3	3024	3025
V@PEOF	001	0002	2998	
V@PSQ2	001	0014	3001	

TOTAL STATEMENTS IN ERROR IN THIS ASSEMBLY = 0

OL105 I THE CODE LENGTH OF #INSTD IS 5632 DECIMAL.  
OL103 I TOTAL NUMBER OF LIBRARY SECTORS REQUIRED IS 27  
NAME-#INSTD,PACK-R1R1R1,UNIT-R1,RETAIN-P,LIBRARY-R,CATEGORY-000

START ADDRESS	CATEGORY	NAME AND ENTRY	CODE LENGTH HEXADECIMAL DECIMAL
---------------	----------	----------------	------------------------------------

0600	0	#INSTD	1600 5632
------	---	--------	-----------

OL100 I THE TOTAL CORE USED BY #INSTD IS 5632 DECIMAL.  
 OL101 I THE START CONTROL ADDRESS OF THIS MODULE IS 0600.  
 OL104 I TOTAL NUMBER OF LIBRARY SECTORS REQUIRED IS 23  
 NAME-#INSTD,PACK-R1R1R1,UNIT-R1,RETAIN-P,LIBRARY-O

\*

```

1532 * FILE ELEMENT. *
1533 * * THIS ROUTINE OPERATES ON THE FOLLOWING PSEUDO INSTRUCTIONS TO *
1534 * ACCESS AND STACK THE CURRENTLY REFERENCED PROGRAM 'DATA' FILE *
1535 * ELEMENT. *
1536 * * 'DCA' - DEFINE CONSTANT ADDRESS (FORMAT - OP VADR) *
1537 * THE DATA ELEMENT AT VIRTUAL ADDRESS VADR IS DEFINED AS AN *
1538 * ELEMENT IN THE 'DATA' FILE. THE POSITION OF THE ELEMENT *
1539 * IN THE FILE IS DIRECTLY RELATED TO THE POSITION OF THE *
1540 * 'DCA' INSTRUCTION WITH RESPECT TO OTHER 'DCA' INSTRUCTIONS *
1541 * IN THE PROGRAM. *
1542 * * 'DDL' - DEFINE 'DATA' LINKAGE (FORMAT - OP VADR) *
1543 * 'DDL' ALWAYS FOLLOWS A STRING OF 'DCA' INSTRUCTIONS. *
1544 * THE 'DCA' INSTRUCTION BEGINNING AT VADR IS THE NEXT *
1545 * SEQUENTIAL 'DCA' IN THE PROGRAM. WHEN VADR = X'0000', *
1546 * 'DDL' MARKS THE END OF THE 'DATA' FILE. *
1547 * * 'EOP' - END OF PMC PAGE (FORMAT - OP) *
1548 * EACH PSEUDO MACHINE CODE VIRTUAL PAGE IS TERMINATED WITH *
1549 * AT LEAST ONE 'EOP' INSTRUCTION. 'EOP' EXECUTION RESULTS *
1550 * IN CONTROL BEING PASSED TO THE FIRST PSEUDO INSTRUCTION *
1551 * WHICH APPEARS IN THE NEXT SEQUENTIAL VIRTUAL PAGE. *
1552 * * 'DATA' FILE POINTER I$DATA CONTAINS EITHER THE VIRTUAL ADDRESS *
1553 * OF A 'DCA' INSTRUCTION OR THAT OF A 'DDL' OR 'EOP' FOLLOWING A *
1554 * STRING OF 'DCA' INSTRUCTIONS. IN THE LATTER CASE, THE CURRENT *
1555 * 'DCA' INSTRUCTION IS THAT INDICATED BY THE 'DDL' OR 'EOP'. *
1556 * THE ELEMENT REFERENCED BY THE OPERAND OF THE CURRENT 'DCA' *
1557 * INSTRUCTION IS STACKED, AND I$DATA IS INCREMENTED TO REFERENCE *
1558 * THE NEXT 'DCA' INSTRUCTION. *
1559 * *
1560 *ENTRY POINTS *
1561 * THIS ROUTINE HAS A SINGLE ENTRY POINT - FZREAD - WHOSE FUNCTION *
1562 * IS DEFINED ABOVE. CALLING SEQUENCE IS *
1563 * B I$CALL *
1564 * DC AL2(V$XS?O) *
1565 * WHERE THE ADDRESS CONSTANT PARAMETER DEFINES THE VIRTUAL ADDRESS *
1566 * OF ENTRY POINT FZREAD. EXECUTION IS SUBJECT TO INPUT CONDITIONS *
1567 * DESCRIBED BELOW. *
1568 * *
1569 *INPUT *
1570 * * I$STAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER. THIS IS TO *
1571 * CONTAIN THE CORE ADDRESS OF THE FIRST AVAILABLE STACK LOCATION. *
1572 * * I$DATA - 2 BYTES, FOR THE 'DATA' FILE POINTER. THIS IS TO *
1573 * CONTAIN THE VIRTUAL ADDRESS OF THE CURRENT 'DCA' INSTRUCTION *
1574 * OR THAT OF A 'DDL' OR 'EOP' INDICATING THE 'DCA' INSTRUCTION. *
1575 * * PMC 'DATA' FILE - 'DATA' FILE ELEMENT-REFERENCING PSEUDO *

```

FZREAD - S/3 BASIC INTERPRETER STATEMENT EXEC RTN						
ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21	PAGE 209
		1576	*	INSTRUCTIONS GENERATED FOR EACH 'DATA' FILE STATEMENT AND		*
		1577	*	ACCESSED USING FILE POINTER I\$DATA.		*
		1578	*			*
		1579	*	*OUTPUT		*
		1580	*	* I\$STAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER, WHEN NO		*
		1581	*	ERROR OCCURS, THIS CONTAINS THE CORE ADDRESS OF THE LEFTMOST		*
		1582	*	BYTE OF THE 'DATA' FILE ELEMENT STACKED DURING FZREAD EXECUTION.		*
		1583	*	* I\$DATA - 2 BYTES, FOR THE 'DATA' FILE POINTER. WHEN NO ERROR		*
		1584	*	OCCURS, THIS CONTAINS THE VIRTUAL ADDRESS OF THE NEXT SEQUEN-		*
		1585	*	TIAL 'DCA' INSTRUCTION OR THAT OF A 'DDL' OR 'EOP' INDICATING		*
		1586	*	THE NEXT 'DCA' INSTRUCTION.		*
		1587	*	* I\$ERRC - 1 BYTE, FOR THE ERROR CONDITION CODE. THIS CONTAINS		*
		1588	*	A NULL CODE (I@NERR) WHEN NO ERROR CONDITION EXISTS OR AN		*
		1589	*	ERROR CODE SPECIFYING THE PARTICULAR ERROR CONDITION DISCOVERED.		*
		1590	*	* RUN-TIME STACK - WHEN NO ERROR CONDITION OCCURS, THIS CONTAINS		*
		1591	*	THE CURRENT 'DATA' FILE ELEMENT AT THE TOP STACK POSITION.		*
		1592	*			*
		1593	*	*EXTERNAL REFERENCES		*
		1594	*	* I\$STCK - ENTRY POINT FOR INTERPRETER ELEMENT STACKING ROUTINE.		*
		1595	*	* ISLDYR - ENTRY POINT FOR PAGING MODLLE CONVERT AND LOAD @XR RTN.		*
		1596	*	* I\$QTRN - ENTRY POINT FOR PAGING MODLLE V.M. TETURN CONTROL RTN.		*
		1597	*	* I\$STAK - 2 BYTES, FOR THE STACK POINTER.		*
		1598	*	* I\$DATA - 2 BYTES, FOR THE PROGRAM 'DATA' FILE POINTER.		*
		1599	*	* I\$VADR - 2 BYTES, FOR PAGING MODULE VIRTUAL ADDRESS PARAMETER.		*
		1600	*	* I\$SLN5 - 1 BYTE, FOR ELEMENT STACKING LENGTH PARAM TO ISTACK.		*
		1601	*	* I\$ERRC - 1 BYTE, FOR THE INTERPRETER EXECUTION ERROR CODE.		*
		1602	*			*
		1603	*	*EXITS, NORMAL		*
		1604	*	CONTROL IS ALWAYS PASSED TO THE PAGING ROUTINE AT ENTRY POINT		*
		1605	*	I\$RTRN (IPGRTN) FOR A RETURN TO CALLING PROGRAM.		*
		1606	*			*
		1607	*	*EXITS, ERROR		*
		1608	*	CONTROL IS PASSED TO THE PAGING ROUTINE AT ENTRY POINT I\$RTRN		*
		1609	*	(IPFRTN) WITH THE PARAMETER I\$ERRC CONTAINING THE APPROPRIATE		*
		1610	*	ERROR MESSAGE CODE.		*
		1611	*			*
		1612	*	*TABLES/WORK AREAS		*
		1613	*	FZREAD PMC EXECUTION BRANCH ADDRESS TABLE - 6 BYTES, FOR 'DATA'		*
		1614	*	FILE DEFINITION PMC OPCODE TRANSLATION TO AN FZREAD ENTRY POINT		*
		1615	*	ADDRESS. THIS TABLE CONSITS OF THREE 2 BYTE ENTRIES CONTAINING		*
		1616	*	THE FOLLOWING INFORMATION -		*
		1617	*	* BYTE 0 - DUMMY SPACER.		*
		1618	*	* BYTE 1 - PAGE DISPLACEMENT WITHIN FZREAD FOR THE INTERNAL		*
		1619	*	ENTRY POINT ASSOCIATED WITH A 'DCA', 'DDL' OR 'EOP' PSEUDO		*
		1620	*	INSTRUCTION.		*
		1621	*			*
		1622	*	*ATTRIBUTES		*
		1623	*	* REUSABLE		*
		1624	*	* NATURALLY RELOCATBLE		*
		1625	*			*
		1626	*	*CHARACTER CODE DEPENDENCY		*
		1627	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR		*
		1628	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.		*
		1629	*			*

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	31/05/21	PAGE 210
		1632	*	* ERROR 1 - FILE POINTER CONTAINS AN INVALID 'DATA'			*
		1633	*	FILE VIRTUAL ADDRESS. AN ERROR CODE FOR THE MESSAGE			*
		1634	*	'NO DATA STATEMENT SPECIFIED IS ESTABLISHED IN INTERPRETER			*
		1635	*	PARAMETER I\$ERRC.			*
		1636	*	* ERROR 2 - A 'DDL' INSTRUCTION WITH OPERAND X'0000' IS EN-			*
		1637	*	COUNTERED WHILE ATTEMPTING TO ACCESS THE NEXT 'DCA' INSTRU-			*
		1638	*	TION. AN ERROR CODE FOR THE MESSAGE 'INSUFFICIENT DATA FOR			*
		1639	*	READ' IS ESTABLISHED IN INTERPRETER PARAMETER I\$ERRC.			*
		1640	*	* IN EACH OF THESE CASES, CONTROL IS PASSED IMMEDIATELY TO			*
		1641	*	PAGING MODULE ENTRY POINT I\$RTRN (IPGRTN).			*
		1642	*				*
		1643	*	REGISTER USAGE			*
		1644	*	* REGISTER @BR IS TO CONTAIN THE CORE PAGE BASE ADDRESS			*
		1645	*	ESTABLISHED THROUGH PAGING MODULE CONTROL FOR THE PAGE WHICH			*
		1646	*	INCLUDES FZREAD, AND IS RESTORED THROUGH THE PAGING MODULE.			*
		1647	*	* RESISTER @XR IS NOT SAVED. IT IS USED IN FZREAD FOR GENERAL			*
		1648	*	PURPOSE INDEXING OPERATIONS.			*
		1649	*				*
		1650	*	SAVED/RESTORED AREAS			*
		1651	*	NONE			*
		1652	*				*
		1653	*	MODIFICATION CONSIDERATIONS			*
		1654	*	'DATA' FILE ELEMENT REFERENCING PMC OPERATION IS BASED UPON			*
		1655	*	THE SEQUENCE AND LENGTH OF THE ENTRIES IN THE FZREAD PSEUDO			*
		1656	*	INSTRUCTION BRANCH ADDRESS TABLE. TABLE ENTRIES ARE SELECTED			*
		1657	*	USING THE NUMERIC REPRESENTATION OF OPCODE 'EOP' AS A BASE			*
		1658	*	DISPLACEMENT, AND ANY CHANGES TO THE RELATIONSHIP BETWEEN THE			*
		1659	*	CONSTANTS FOR ALL OPCODES OPERATED ON BY THIS ROUTINE MUST			*
		1660	*	TAKE FULL CONSIDERATIONS OF THIS TABLE USAGE AND ORGANIZATION.			*
		1661	*				*
		1662	*	REQUIRED MODULES			*
		1663	*	* @SYSEQ - COMMON SYSTEM EQUATES.			*
		1664	*	* @ERMEQ - SYSTEM ERROR MESSAGE CODE EQUATES			*
		1665	*	* \$B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.			*
		1666	*	* \$I\$EQU - INTERPRETER FIXED LOCATION ADDRESS EQUATES.			*
		1667	*	* \$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD. PREC, ONLY)			*
		1668	*	* \$I@LEQ - INTERPRETER PARAMETER EQUATES (FOR LONG PREC, ONLY)			*
		1669	*				*
		1670	*	OTHER			*
		1671	*	NONE			*
		1672	*	*****			*

## FZREAD - S/3 BASIC INTERPRETER STATEMENT EXEC RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 211
		1674		*****	
		1675		* START OF READ STATEMENT EXECUTION MODULE	*
		1676		*****	
		1677		*	
		1678		* ESTABLISH ADDRESSABILITY FOR THE READ STATEMENT ROUTINE	
		1679		*	
3300		1680		ORG *,B@LVPG,0	BEGIN AT PAGE BOUNDARY
		1681	3300	USING *,@BR	DEFINE READ RTN BASE ADDRESS
		1682		*	
		1683		* ENTER FZREAD - TEST FOR A DATA STATEMENT SPECIFICATION.	
		1684		*	
		1685	3300	FZREAD EQU *	FZREAD ENTRY POINT
		1686		*	
3300 3D 56 0D52		1687		CLI I\$DATA-1,@VENTA	IF DATA POINTER IS DEFINED
3304 F2 02 08		1688		JNL FZR020	* GO CONTINUE 'READ' EXECUTION
		1689		*	
		1690		* NO DATA STATEMENT - SET 'NO DATA STATEMENT SPECIFIED' ERROR MESSAGE	
		1691		*	
3307 3C BE 0CBC		1692		FZR010 MVI I\$ERRC,@E720	SET INTERPRETER ERROR CODE
330B C0 87 12D3		1693		B I\$RTRN	RETURN TO TERMINATE EXECUTION
		1694		*	
		1695		* LOAD THE DATA PMC PAGE INTO CORE VIRTUAL MEMORY - THIS PAGE CONTAINS	
		1696		* (IN GENERAL) A SERIES OF 'DCA' INSTRUCTIONS WHICH DEFINE THE VADDR'S	
		1697		* OF THE CONSTANTS WHICH COMPRISE THE PROGRAM DATA FILE.	
		1698		*	
330F 4C 01 19 0D53		1699		FZR020 MVC FZR030(,@BR),I\$DATA(@VADDR)	SET PAGING PARAMETER TO LOAD
		1700		*	* CURRENT DATA FILE OPCODE
3314 C0 87 1330		1701		B I\$LDXR	LINK TO LOAD CURR DATA FILE PMC
3318		1702	3319	FZR030 DS CL(@VADDR)	VADDR OF CURR DATA FILE OPCODE
		1703		*	
		1704		* ESTABLISH BRANCH ADDRESS FROM OPCODE DISPLACEMENT TABLE	
		1705		*	
331A 74 02 2B		1706		FZR040 ST FZR060+@OP1(,@BR),@XR	SAVE THE DATA FILE OPCODE CADDR
331D 6C 00 27 00		1707		MVC FZR050+@DD2(,@BR),I@XOPC(B@LCOP,@XR)	MOVE OPCODE TO DISP
3321 D2 02 06		1708		LA FZRBAT-B@CEOP+1(,@BR),@XR	LOAD BRANCH TABLE BASE ADDR
3324 6C 00 2E 00		1709		FZR050 MVC FZR070+@D1(,@BR),*-(1,@XR)	MOVE TABLE ENTRY TO BR INST
3328 C2 02 0000		1710		FZR060 LA *-*,@XR	RESTORE DATA FILE OPCODE CADDR
		1711		*	
		1712		* BRANCH TO EXECUTION ROUTINE SPECIFIED BY THE DATA FILE OPCODE	
		1713		*	
332C D0 87 00		1714		FZR070 B *-*(,@BR)	GO EXECUTE CURR DATA FILE PMC
		1715		*	
		1716		*****	

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 212
				1718		*****				
				1719	*	F7RDCA - DEFINE VIRTUAL ADDRESS OR CURRENT DATA FILE ELEMENT	*			
				1720		*****				
				1721	*					
		332F		1722	FZRDCA EQU	*	BEGIN FZRDCA EXECUTION			
				1723	*					
				1724	*	STACK THE DATA ELEMENT SPECIFIED BY THE 'DCA' VIRTUAL ADDRESS OPERAND				
				1725	*					
332F	2C	01	144A 02	1726	FZR080 MVC	I\$VADR,I@XVAD(B@LCVA,@XR)	SET PAGING PARAM FOR DATA VADDR			
3334	3C	12	0BA2	1727		MVI I\$SLNG,I@LCRV-1	SET STACKING ROUTINE TO STACK			
				1728	*		* MAXIMUM LENGTH DATA ELEMENT			
3338	35	02	0D4E	1729		L I\$STAK,@XR	LOAD THE STACK POINTER			
333C	C0	87	0B50	1730		B I\$STCK	LINK TO STACK THE DATA ELEMENT			
				1731	*					
				1732	*	ADVANCE DATA FILE POINTER TO REFERENCE NEXT DATA FILE PMC				
				1733	*					
3340	1E	00	0D53 6C	1734	FZR090 ALC	I\$DATA,FZRLDA(@VADDR-1,@BR)	INCREMENT DATA FILE POINTER			
3345	C0	87	12D3	1735		B I\$RTRN	RETURN TO THE INTERPRETER			
				1736	*					
				1737		*****				

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 213
		1739		*****	
		1740	*	FZRDDL - DEFINE LINKAGE ADDRESS FOR NEXT DATA FILE PSEUDO INSTR.	*
		1741		*****	
		1742	*		
		3349	1743	FZRDDL EQU *	BEGIN FZRDDL EXECUTION
		1744	*		
		1745	*	TEST FOR END OF THE PROGRAM DATA FILE	
		1746	*		
3349	BD 56 01	1747	FZR100	CLI I@XVAD-1(,@XR),@VENTA	IF 'DDL' OPERAND IS VALID VADDR
334C	F2 02 08	1748		JNL FZR120	* GO PERFORM LINKAGE OPERATION
		1749	*		
		1750	*	END OF DATA FILE - SET 'INSUFFICIENT DATA FOR READ' ERROR MESSAGE	
		1751	*		
334F	3C BF 0CBC	1752	FZR110	MVI I\$ERRC,@E721	SET INTERPRETER ERROR CODE
3353	C0 87 12D3	1753		B I\$RTRN	RETURN TO TERMINATE EXECUTION
		1754	*		
		1755	*	DATA FILE CONTINUED - LINK TO NEXT DATA FILE PMC SEQUENCE	
		1756	*		
3357	2C 01 0D53 02	1757	FZR120	MVC I\$DATA,I@XVAD(B@LCVA,@XR)	SET DATA FILE PT - LINKAGE ADDR
335C	D0 87 0F	1758		B FZR020(,@BR)	GO PROCESS NEXT DATA FILE PMC
		1759	*		
		1760		*****	



ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 214
				1762		*****				
				1763	*	FZREOP - CONTINLE DATA FILE PMC ON NEXT VIRTUAL PAGE				*
				1764		*****				
				1765	*					
				335F 1766	FZREOP EQU	*	BEGIN FZREOP EXECUTION			
				1767	*					
				1768	*	ADVANCE DATA FILE POINTER TO REFERENCE 1ST PSUEDO INSTRUCTION ON				
				1769	*	NEXT SEQUENTIAL VIRTUAL PAGE.				
				1770	*					
335F	1E	00	0D52 6B	1771	FZR130	ALC	I\$DATA-1,FZRBNI(1,@BR)	INCREMENT	POINTER	PAGE NUMBER
3364	3C	00	0D53	1772		MVI	I\$DATA,@ZERO	SET	POINTER	PAGE DISP TO ZERO
3368	D0	87	0F	1773		B	FZR020(,@BR)	GO	PROCESS	NEXT DATA FILE PMC
				1774	*					
				1775		*****				

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 215
			1777	*****				
			1778	* READ STATEMENT EXECUTION ROUTINE CONSTANTS				*
			1779	*****				
			1780	*				
336B 01		336B	1781	FZRBN1 DC	IL1'1' BINARY INTEGER +1			
336C 03		336C	1782	FZRLDA DC	AL1(B@LDCA) LENGTH OF 'DCA' PSEUDO INST			
			1783	*				
			1784	*****				
			1785	* READ STMT RTN PSEUDO OPCODE EXECUTION BRANCH ADDRESS TABLE				
			1786	*****				
			1787	*				
		336D	1788	FZRBAT EQU *	BRACH TABLE STARTING ADDRESS			
			1789	*				
336D 005F		336E	1790	DC	AL(@CADDR)(FZREOP-FZREAD) EOP (X'68') END OF PMC PAGE			
336F 002F		3370	1791	DC	AL(@CADDR)(FZRDCA-FZREAD) DCA (X'6A') DEFINE CON VADDR			
3371 0049		3372	1792	DC	AL(@CADDR)(FZRDDL-FZREAD) DDL (X'6C') DEFINE DATA LINK			
			1793	*				
			1794	*****				
			1795	*				
			1796	* END OF READ STATEMENT EXECUTION ROUTINE CODING				

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 216

```

1798 *****
1799 * 5703-XM1 COPYRIGHT IBM CORP. 1970 *
1800 * REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083 *
1801 * *
1802 *****
1803 *STATUS *
1804 * VERSION 1 MODIFICATION 0 *
1805 * *
1806 *FUNCTION - *
1807 * * FZSPRT EXECUTION CAUSES DATA OUTPUT AND/OR CARRIER/CURSOR *
1808 * POSITIONING ON THE SYSTEM PRINT DEVICE UNDER CONTROL OF CODES *
1809 * DEVELOPED FROM THE FORMAT SPECIFIED IN A BASIC PROGRAM 'PRINT' *
1810 * STATEMENT. *
1811 * * THE FOLLOWING ACTIONS ARE PERFORMED, DEPENDING ON THE CODE *
1812 * STORED IN INTERPRETER PARAMETER I$PARM - *
1813 * * CODE X'01' - PRINT AND NO SPACE. *
1814 * THE DATA ELEMENT AT THE TOP OF THE RUN?TIME STACK IS CON- *
1815 * VERTED TO OUTPUT FORMAT AND PRINTED. IF THE ELEMENT IS *
1816 * ARITHMETIC, THE CARRIER/CURSOR IS RETURNED TO THE START OF *
1817 * THE NEXT LINE (BEFORE PRINTING) WHEN THE CURRENT LINE CAN- *
1818 * NOT CONTAIN THE FORMATTED VALUE. THE CARRIER/CURSOR IS *
1819 * LEFT POSITIONED AT THE END OF THE PRINTED VALUE. *
1820 * * CODE X'02' - PRINT AND SPACE FULL ZONE. *
1821 * THE DATA ELEMENT AT THE TOP OF THE RUN-TIME STACK IS CON- *
1822 * VERTED TO OUTPUT FORMAT AND PRINTED. IF THE ELEMENT IS *
1823 * ARITHMETIC, THE CARRIER/CURSOR IS RETURNED TO THE START OF *
1824 * THE NEXT LINE (BEFORE PRINTING) WHEN THE CURRENT LINE CAN- *
1825 * NOT CONTAIN THE FORMATTED VALUE. IF THE ELEMENT IS A *
1826 * CHARACTER REFERENCE, THE CARRIER/CURSOR IS RETURNED TO THE *
1827 * START OF THE NEXT LINE (BEFORE PRINTING) WHEN THE CURRENT *
1828 * LINE DOES NOT CONTAIN A FULL PRINT ZONE (18 SPACES). AT *
1829 * THE END OF PRINTING, THE CARRIER/CURSOR IS SPACED TO THE *
1830 * END OF THE FULL PRINT ZONE. *
1831 * * CODE X'03' - PRINT AND SPACE PACKED ZONE. *
1832 * THE DATA ELEMENT AT THE TOP OF THE RUN-TIME STACK IS CON- *
1833 * VERTED TO OUTPUT FORMAT AND PRINTED. IF THE ELEMENT IS *
1834 * ARITHMETIC, THE CARRIER/CURSOR IS RETURNED TO THE START OF *
1835 * THE NEXT LINE (BEFORE PRINTING) WHEN THE CURRENT LINE CAN *
1836 * NOT CONTAIN THE FORMATTED VALUE. AFTER AN ARITHMETIC ELE- *
1837 * MENT IS PRINTED, THE CARRIER/CURSOR IS SPACED TO THE END *
1838 * OF THE PACKED PRINT ZONE DEFINED IN FUNCTIONAL SPECIF1- *
1839 * CATIONS. AFTER A CHARACTER ELEMENT IS PRINTED, THE *
1840 * CARRIER/CURSOR IS LEFT POSITIONED AT THE END OF THE *
1841 * PRINTED ELEMENT. *
1842 * * CODE X'04' - PRINT AND RETURN CARRIER/CURSOR. *
1843 * THE DATA ELEMENT AT THE TOP OF THE RUN-TIME STACK IS CON- *
1844 * VERTED TO OUTPUT FORMAT AND PRINTED. IF THE ELEMENT IS *
1845 * ARITHMETIC, THE CARRIER/CURSOR IS RETURNED TO THE START OF *
1846 * THE NEXT LINE (BEFORE PRINTING) WHEN THE CURRENT LINE CAN- *
1847 * NOT CONTAIN THE FORMATTED VALUE. AFTER THE ELEMENT IS *
1848 * PRINTED, THE CARRIER/CURSOR IS RETURNED TO THE START OF *
1849 * THE NEXT LINE. *
1850 * * CODE X'05' - SPACE FULL ZONE. *
1851 * THE CARRIER/CURSOR IS SPACED 18 CHARACTERS. IF NO MORE *
1852 * THAN 18 CHARACTERS REMAIN IN THE CURRENT LINE, THE *
1853 * CARRIER/CURSOR IS RETURNED TO THE START OF THE NEXT LINE. *

```

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 217
		1854	*	* CODE X'06' - SPACE PACKED ZONE.	*
		1855	*	THE CARRIER/CURSOR IS SPACED 3 CHARACTERS, IF NO MORE	*
		1856	*	THAN 3 CHARACTERS REMAIN IN THE CURRENT LINE, THE	*
		1857	*	CARRIER/CURSOR IS RETURNED TO THE START OF THE NEXT LINE.	*
		1858	*	* CODE X'07' - RETURN CARRIER/CURSOR,	*
		1859	*	THE CARRIER/CURSOR IS RETURNED. TO THE START OF THE NEXT	*
		1860	*	LINE.	*
		1861	*	* CODE X'08' - RETURN CARRIER/CURSOR ON CONDITION.	*
		1862	*	WHEN THE CURRENT LINE DOES NOT CONTAIN MORE THAN 18 CHAR-	*
		1863	*	ACTERS, THE CARRIER/CURSOR IS RETURNED TO THE START OF THE	*
		1864	*	NEXT LINE.	*
		1865	*	* WHEN REQUIRED, ELEMENT CONVERSION AND OUTPUT ARE PERFORMED IN	*
		1866	*	THE RUN-TIME STACK, SO TWAT THE STACKED ELEMENT IS NOT RECOVER-	*
		1867	*	ABLE. AFTER PRINTING, ARITHMETIC ELEMENT OUTPUT FORMAT DEPENDS	*
		1868	*	ON THE MAGNITUDE AND FRACTIONAL CHARACTERISTICS OF THE VALUE.	*
		1869	*	CHARACTER REFERENCE FORMATTING INVOLVES TRUNCATION OF TRAILING	*
		1870	*	BLANKS. CHARACTER CONSTANTS (LITERALS) ARE PRINTED AS SPECI-	*
		1871	*	FIED IN THE 'PRINT' STATEMENT.	*
		1872	*	* EITHER THE MATRIX PRINTER OR THE CRT (OR BOTH) MAY BE USED FOR	*
		1873	*	OUTPUT, DEPENDING ON THE CURRENT DEFINITION OF THE SYSTEM PRINT	*
		1874	*	DEVICE. CRT OUTPUT IS BASED ON A FIXED DISPLAY WIDTH OF 64	*
		1875	*	CHARACTERS, WHILE PRINTER LINE WIDTH IS BASED ON THAT ASSIGNED	*
		1876	*	THROUGH THE 'WIDTH' SYSTEM COMMAND.	*
		1877	*		*
		1878	*	*ENTRY POINTS	*
		1879	*	THIS ROUTINE HAS A SINGLE ENTRY POINT - FZSPRT - WHOSE FUNCTION	*
		1880	*	IS DEFINED ABOVE. CALLING SEQUENCE IS -	*
		1881	*	B I\$CALL	*
		1882	*	DC AL2(V\$XSPR)	*
		1883	*	WHERE THE ADDRESS CONSTANT PARAMETER DEFINES THE VIRTUAL ADDRESS	*
		1884	*	OF ENTRY POINT FZSPRT. EXECUTION IS SUBJECT TO INPUT CONDITIONS	*
		1885	*	DESCRIBED BELOW.	*
		1886	*		*
		1887	*	*INPUT	*
		1888	*	* #ISPARM - 2 BYTES, FOR THE PRINT CONTROL PARAMETER. THIS CON-	*
		1889	*	TAINS A CONTROL CODE, AS INDICATED UNDER 'FUNCTION', IN THE	*
		1890	*	RIGHTMOST BYTE.	*
		1891	*	* I\$STAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER. FOR THOSE	*
		1892	*	CONTROL CODES SPECIFYING A DATA ELEMENT (SEE 'FUNCTION') THIS	*
		1893	*	CONTAINS, THE CORE ADDR OF THE FIRST AVAILABLE STACK LOCATION.	*
		1894	*	* RUN-TIME STACK - THIS CONTAINS AN UNPACKED FLOATING POINT VALUE	*
		1895	*	OR CHARACTER ELEMENT IN THE TOP STACK POSITION FOR CONTROL	*
		1896	*	CODES SPECIFYING DATA OUTPUT (SEE 'FUNCTION').	*
		1897	*	* I\$SLLC - 1 BYTE, FOR THE LENGTH CODE DEFINING THE LAST STACKED	*
		1898	*	DATA ELEMENT. WHEN DATA OUTPUT IS SPECIFIED, THIS IS USED TO	*
		1899	*	DETERMINE THE TYPE OF DATA ITEM (ARITHMETIC OR CHARACTER) CON-	*
		1900	*	TAINED IN THE TOP STACK POSITION.	*
		1901	*	* \$PRPOS - 1 BYTE, FOR THE MATRIX PRINTER CARRIER POSITION	*
		1902	*	INDICATORS. THIS CONTAINS THE CARRIER POSITION, RELATIVE TO	*
		1903	*	THE HARDWARE LEFT MARGIN AS 0, OF THE MATRIX PRINTER CARRIER.	*
		1904	*	* \$RMRGN - 1 BYTE, FOR THE MATRIX PRINTER SOFTWARE RIGHT MARGIN	*
		1905	*	INDICATOR.	*
		1906	*	* \$CRPOS - 1 BYTE, FOR THE CRT CURSOR POSITION INDICATOR. THIS	*
		1907	*	CONTAINS THE CURSOR POSITION, RELATIVE TO THE LEFT CRT MARGIN	*
		1908	*	AS 0, OF THE CRT CURSOR.	*
		1909	*	* \$PRDEV - 2 BYTES, FOR THE SYSTEM PRINT DEVICE INDICATOR.	*

FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN					
ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 218
		1910	*	* \$EXFTR - 1 BYTE, FOR THE SYSTEM CORE EXTENSION FACTOR.	*
		1911	*		*
		1912	*	*OUTPUT	*
		1913	*	* PRINTED OUTPUT AND/OR CARRIER/CURSOR CONTROL - AS SPECIFIED BY	*
		1914	*	THE CODE IN I\$PARM, THE TYPE OF DATA ELEMENT IN THE STACK, AND	*
		1915	*	THE CURRENTLY DEFINED SYSTEM PRINT DEVICE(S).	*
		1916	*	* I\$PARM - 2 BYTES, FOR THE PRINT CONTROL PARAMETER, THIS INPUT	*
		1917	*	CONTROL CODE IS DESTROYED DURING EXECUTION.	*
		1918	*	* RUN-TIME STACK - WHEN A DATA ELEMENT HAS BEEN PRINTED, THE	*
		1919	*	STACKED ELEMENT HAS BEEN CONVERTED IN PLACE TO OUTPUT FORMAT.	*
		1920	*	* \$PRPOS - 1 BYTE, FOR THE MATRIX PRINTER CARRIER POSITION	*
		1921	*	INDICATOR. THIS HAS BEEN MODIFIED TO INDICATE THE CURRENT	*
		1922	*	CARRIER POSITION AFTER PRINTED OUTPUT WHEN THE MATRIX PRINTER	*
		1923	*	IS A SYSTEM PRINT DEVICE.	*
		1924	*	* \$CRPOS - 1 BYTE, FOR THE CRT CURSOR POSITION INDICATOR. THIS	*
		1925	*	HAS BEEN MODIFIED TO INDICATE CURRENT CURSOR POSITION AFTER	*
		1926	*	DISPLAYED OUTPUT WHEN THE CRT IS A SYSTEM PRINT DEVICE.	*
		1927	*		*
		1928	*	*EXTERNAL REFERENCES	*
		1929	*	* VSSPRT - VIRTUAL ENTRY ADDRESS FOR DFPRNT, V.M. MATRIX PRT IOCS.	*
		1930	*	* DSPLYN - ENTRY POINT FOR THE SYSTEM CRT IOCS (LABEL DSPLYN IS	*
		1931	*	REFERENCED INDIRECTLY USING I\$CSXA TO BUILD A CODE ADDRESS).	*
		1932	*	* I\$CALL - ENTRY POINT FOR PAGING MODULE V.M. PROGRAM CALL RTN.	*
		1933	*	* I\$RTRN - ENTRY POINT FOR PAGING MODULE V.M. RETURN CONTROL RTN.	*
		1934	*	* I\$CSXA - CORE ADDRESS OF 1ST BYTE IN CORE EXTENSION PAST 8K.	*
		1935	*	* I\$PARM - 2 BYTES, FOR THE INTERPRETER COMMUNICATIONS PARAMETER.	*
		1936	*	* I\$STAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER.	*
		1937	*	* I\$SLLC - 1 BYTE, FOR LENGTH CODE (L-1) OF LAST STACKED ELEMENT.	*
		1938	*	* I\$WRK1 - 2 BYTES, FOR INTERPRETER COMMON WORK AREA 1.	*
		1939	*	* I\$WRK2 - 2 BYTES, FOR INTERPRETER COMMON WORK AREA 2.	*
		1940	*	* \$PRPOS - 1 BYTE, FOR MATRIX PRINTER CARRIER POSITION INDICATOR.	*
		1941	*	* \$RMRGN - 1 BYTE, FOR POSITION OF SOFTWARE RIGHT PRINTER MARGIN.	*
		1942	*	* \$CRPOS - 1 BYTE, FOR CRT CURSOR POSITION INDICATOR.	*
		1943	*	* \$PRDEV - 2 BYTES, FOR THE SYSTEM PRINT DEVICE INDICATOR.	*
		1944	*	* \$EXFTR - 1 BYTE, FOR THE SYSTEM CORE EXTENSION FACTOR.	*
		1945	*		*
		1946	*	*EXITS, NORMAL	*
		1947	*	CONTROL IS ALWAYS PASSED TO THE PAGING ROUTINE AT ENTRY POINT	*
		1948	*	I\$RTRN (IPGRTN) FOR A RETURN TO THE CALLING PROGRAM.	*
		1949	*		*
		1950	*	*EXITS, ERROR	*
		1951	*	N/A	*
		1952	*		*
		1953	*	*TABLES/WORKAREAS	*
		1954	*	* FZSPRT BRANCH DISPLACEMENT TABLE - USED TO DIRECT OUTPUT OPERA-	*
		1955	*	TIONS FOR SPECIFIC ELEMENT TYPE - CONTROL CODE COMBINATIONS.	*
		1956	*	* NUMBER OF TABLE ENTRIES - 16	*
		1957	*	* TABLE ENTRY LENGTH - 1 BYTE	*
		1958	*	* ENTRY FORMAT - SINGLE BYTE DISPLACEMENT WITHIN AN FZSPRT	*
		1959	*	VIRTUAL PAGE FOR THE INTERNAL ENTRY POINT ASSOCIATED WITH	*
		1960	*	EACH ELEMENT-CONTROL COMBINATION.	*
		1961	*	* RUN-TIME STACK - THE FIRST 20 AVAILABLE STACK LOCATIONS	*
		1962	*	(INCLUDING LOCATIONS CONTAINING AN ELEMENT TO BE CONVERTED) ARE	*
		1963	*	USED AS THE 'PRINT' OUTPUT BUFFER.	*

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 219
		1966	*	* REUSABLE	*
		1967	*	* NATURALLY RELOCATABLE	*
		1968	*		*
		1969	*	*CHARACTER CODE DEPENDENCY	*
		1970	*	OPERATION OR THIS MODULE DEPENDS UPON THE FOLLOWING PROPER-	*
		1971	*	TIES QF THE INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
		1972	*	* MOST CODING HAS BEEN ARRANGED SO THAT REDEFINITION OF CHAR-	*
		1973	*	ACTER CONSTANTS, BY REASSEMBLY, WILL RESULT IN A CORRECT	*
		1974	*	MODULE FOR THE NEW DEFINITION.	*
		1975	*	* NUMERIC CHARACTERS 0 THROUGH 9 ARE PRESUMED TO BE CODED SUCH	*
		1976	*	THAT THE HIGH ORDER FOUR BITS CONTAIN A SIGN ZONE WITH X'F'	*
		1977	*	DEFINING A POSITIVE DIGIT.	*
		1978	*	THE SPECIFIC INSTRUCTIONS (INSTRUCTION SEQUENCES) WHICH REQUIRE	*
		1979	*	MODIFICATION IF THESE PROPERTIES OF THE CHARACTER SET ARE CHANGED	*
		1980	*	MAY OF IDENTIFIED BY -	*
		1981	*	* THE 4 INSTRUCTIONS BEGINNING AT LABEL FZS035.	*
		1982	*	* THE SINGLE INSTRUCTION IDENTIFIED BY LABEL FZS410.	*
		1983	*	* THE SINGLE INSTRUCTION IDENTIFIED BY LABEL FZS435.	*
		1984	*		*
		1985	*	*NOTES	*
		1986	*	ERROR PROCEDURES	*
		1987	*	FZSPRT UTILIZES OUTPUT IOCS ROUTINES DFPRNT (MATRIX PRINTER)	*
		1988	*	AND DSPLYN (CRT), AND IS SUBJECT TO THE ERP'S INHERENT IN	*
		1989	*	THESE PROGRAMS. FZSPRT OTHERWISE CONTAINS NO ERROR CONDITION	*
		1990	*	TESTS.	*
		1991	*		*
		1992	*	REGISTER USAGE	*
		1993	*	* REGISTER @BR IS TO CONTAIN THE CORE PAGE BASE ADDRESS	*
		1994	*	ESTABLISHED THROUGH PAGING MODULE CONTROL FOR THE PAGE WHICH	*
		1995	*	INCLUDES FZSPRT, AND IS RESTORED THROUGH THE PAGING MODULE.	*
		1996	*	* REGISTER @XR IS NOT SAVED, IT IS USED IN FZSPRT FOR GENERAL	*
		1997	*	PURPOSE INDEXING OPERATIONS.	*
		1998	*		*
		1999	*	SAVED/RESTORED AREAS	*
		2000	*	NONE	*
		2001	*		*
		2002	*	MODIFICATION CONSIDERATIONS	*
		2003	*	NONE	*
		2004	*		*
		2005	*	REQUIRED MODULES	*
		2006	*	* @SYSEQ - COMMON SYSTEM EQUATES.	*
		2007	*	* @FXDEQ - SYSTEM NUCLEUS ADDRESSES AND INDICATOR EQUATES.	*
		2008	*	* \$V\$EQU - VIRTUAL MEMORY FIXED ADDRESS EQUATES.	*
		2009	*	* \$B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.	*
		2010	*	* \$I@EQU - INTERPRETER FIXED LOCATION ADDRESS EQUATES.	*
		2011	*	* \$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD. PREC. ONLY).	*
		2012	*	* \$I@LEQ - INTERPRETER PARAMETER EQUATES (FOR LONG PREC. ONLY).	*
		2013	*		*
		2014	*	OTHER	*
		2015	*	NONE	*
		2016	*	*****	*

## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 220
		2018		*****	
		2019		* START OF PRINT STATEMENT EXECUTION MODULE	*
		2020		*****	
		2021		*	
		2022		* ESTABLISH ADDRESSABILITY FOR PRINT ROUTINE 1ST VM PAGE	
		2023		*	
		2024		*FZSP1B VPAGE 0	
3400		2025		ORG *,256,0	SET STARTING ADDRESS
		2026	3400	FZSP1B EQU *	START OF PROGRAM CODING
3301		2027		ORG *-255	RESET IAR TO PAGE
3400		2028		ORG *,256,0	* BOUNDARY ADDRESS
		2029	3400	USING *,@BR	SET PAGE BASE ADDRESS
3400		2030		ORG FZSP1B	RESET STARTING ADDRESS
		2031		*** END OF EXPANSION ***	
		2032		*	
		2033		* ENTER FZSPRT - ACCESS THE STACKED DATA ELEMENT	
		2034		*	
		2035	3400	FZSPRT EQU *	FZSPRT ENTRY POINT
3400 35 02 0D4E		2036		L I\$STAK,@XR	LOAD THE STACK POINTER
		2037		*	
		2038		* INITIALIZE AND TEST FOR CARRIER CONTROL (ONLY) PARAMETER	
		2039		*	
3404 7C 00 C7		2040		FZS010 MVI FZSCNT(,@BR),@ZERO	CLEAR DATA CHARACTER COUNTER
		2041		*	
3407 3D 05 0D57		2042		CLI I\$PARM,B@PRSL	IF CARRIER CONTROL ONLY,
340B D0 02 A4		2043		BNL FZS180(,@BR)	* GO PERFORM THE OPERATION
		2044		*	
		2045		* TEST FOR CHARACTER ELEMENT PROCESSING	
		2046		*	
340E 3D 12 0BA1		2047		FZS020 CLI I\$SLLC,I@LCRV-1	IF STACK CONTAINS CHAR ELEMENT
3412 D0 81 73		2048		BE FZS130(,@BR)	* GO ESTABLISH CHARACTER OUTPUT
		2049		*	
		2050		*****	



ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 221
				2052			*****			
				2053			* ARITHMETIC ELEMENT CONVERSION TO OUTPUT FORMAT			*
				2054			*****			
				2055			*			
				2056			* PROCESS THE SIGN OF THE STACKED ARITHMETIC VALUE			
				2057			*			
3415	7C	40	6E	2058	FZS030	MVI	FZS120+@Q(,@BR),B@BLNK			SET SIGN CHARACTER TO BLANK
3418	B8	F0	07	2059	FZS035	TBN	I@SIGN(,@XR),B@ZPOS			IF STACKED VALUE IS POSITIVE
341B	F2	10	06	2060		JT	FZS040			* SKIP PAST MINUS PROCESSING
341E	7C	60	6E	2061		MVI	FZS120+@Q(,@BR),B@MINS			SET SIGN CHARACTER TO MINUS
3421	BA	F0	07	2062		SBN	I@SIGN(,@XR),B@ZPOS			MAKE STACKED VALUE POSITIVE
3424	7C	01	C7	2063	FZS040	MVI	FZSCNT(,@BR),@B1			SET CHARACTER COUNT FOR SIGN
				2064			*			
				2065			* TEST FOR A ZERO VALUE (CATEGORIZED AS AN INTEGER) - A ZERO VALUE IS			
				2066			* LEFT IN THE STACK IN THE FORM 'S0', WHERE 'S' IS THE SIGN POSITION			
				2067			*			
3427	BD	F0	01	2068	FZS050	CLI	I@MANL(,@XR),B@DEC0			IF MOST SIGNIFICANT DIGIT NOT
342A	F2	01	07	2069		JNE	FZS060			* ZERO, GO ESTABLISH FORMAT
342D	5E	00	C7 DF	2070		ALC	FZSCNT(,@BR),FZSBN1(1,@BR)			INCR CHAR COUNT FOR ZERO DIGIT
3431	F2	87	39	2071		J	FZS120			* AND GO SET FOR ARITH OUTPUT
				2072			*			
				2073			* VALUE NOT ZERO - TEST MAGNITUDE FOR OUTPUT IN E- OR F-FORMAT			
				2074			*			
3434	BD	81	00	2075	FZS060	CLI	I@DEXP(,@XR),B@NXZR+1			IF VALUE LESS THAN 1E+0, OR
3437	F2	82	28	2076		JL	FZS110			* GREATER THAN OR EQUAL TO
343A	BD	86	00	2077		CLI	I@DEXP(,@XR),B@NXZR+I@APRC			* 1E+6 (1E+11 FOR LONG PREC),
343D	F2	84	22	2078		JH	FZS110			* GO CONVERT TO E OR F FORMAT
				2079			*			
				2080			* POSSIBLE I-FORMAT - TEST FOR A FRACTIONAL COMPONENT			
				2081			*			
3440	6C	00	56 00	2082	FZS070	MVC	FZS090+@Q(,@BR),I@DEXP(1,@XR)			ESTABLISH THE NUMBER OF
3444	5F	00	56 E0	2083		SLC	FZS090+@Q(,@BR),FZSNXZ(1,@BR)			* INTEGER DIGIT POSITIONS
3448	7C	07	4D	2084		MVI	FZS080+@D1(,@BR),I@PREC			SET DISP FOR MANTISSA RH BYTE
				2085			*			
				2086			*			
344B	BD	F0	00	2087	FZS080	CLI	*-*(,@XR),B@DEC0			IF FRACTIONAL DIGIT, GO CONVERT
344E	F2	01	11	2088		JNE	FZS110			* THE VALUE FOR E- OR F-FORMAT
3451	5F	00	4D DF	2089		SLC	FZS080+@D1(,@BR),FZSBN1(1,@BR)			DECR THE MANTISSA POINTER
3455	7D	00	4D	2090	FZS090	CLI	FZS080+@D1(,@BR),*-*			IF MORE FRACTIONAL POSITIONS
3458	D0	84	4B	2091		BH	FZS080(,@BR)			* REMAIN, GO REPEAT LOOP
				2092			*			
				2093			* NO FRACTIONAL COMPONENT - VALUE IS LEFT IN THE STACK IN THE FORM			
				2094			* 'S123' (I-FORMAT) WHERE 'S' IS THE SIGN POSITION			
				2095			*			
345B	5E	00	C7 4D	2096	FZS100	ALC	FZSCNT(,@BR),FZS080+@D1(1,@BR)			INCR CHAR COUNT FOR DIGITS
345F	F2	87	0B	2097		J	FZS120			* AND GO SET FOR ARITH OUTPUT
				2098			*			
				2099			* VALUE CANNOT BE HANDLED USING I-FORMAT - ROUND AND CONVERT VALUE,			
				2100			* LEAVING IN STACK IN THE FORM 'S123.45' (F-FORMAT) OR 'S1.239E+9'			
				2101			* (E-FORMAT) WHERE 'S' IS THE SIGN POSITION.			
				2102			*			
3462	C0	87	12B1	2103	FZS110	B	I\$CALL			LINK TO ROUND AND CONVERT THE
3466	3500			2104		DC	AL(@VADDR)(FZS300)			* VALUE TO E- OR F-FORMAT
				2105			*			
3468	4E	00	C7 0D56	2106		ALC	FZSCNT(,@BR),I\$PARM-1(1)			INCR CHAR COUNT FROM CONVERSION
				2107			*			

[illegible]

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 223
		2116		*****	
		2117		* CHARACTER ELEMENT CONVERSION TO OUTPUT FORMAT	*
		2118		*****	
		2119		*	
		2120		* DETERMINE THE TYPE OF CHARACTER ELEMENT IN THE STACK	
		2121		*	
3473	B8 20 00	2122	FZS130 TBN	I@STAT(,@XR),B@CTYP	IF ELEMENT IS A STRING SEGMENT
3476	F2 10 1C	2123	JT	FZS160	* GO ESTABLISH SEGMENT PARAMS
		2124		*	
		2125		* ELEMENT IS FROM A CHARACTER REFERENCE - LEAVE ELEMENT IN STACK IN	
		2126		* THE FORM 'REFERENCE' (NO TRAILING BLANKS)	
		2127		*	
3479	1E 00 0D57 E1	2128	FZS140 ALC	I\$PARM,FZSCAJ(1,@BR)	ADJUST OUTPUT CONTROL PARAMETER
		2129		*	* FOR CHARACTER REFERENCE
347E	7C 13 8A	2130	MVI	FZS155+@D1(,@BR),I@LCRF+1	SET DISP FOR BYTE AFTER ELEMENT
3481	5F 00 8A DF	2131	FZS150 SLC	FZS155+@D1(,@BR),FZSBN1(1,@BR)	DECR THE ELEMENT POINTER
3485	F2 81 06	2132	JE	FZS158	BRANCH IF ALL CHARS ARE BLANKS
3488	BD 40 00	2133	FZS155 CLI	*-(,@XR),B@BLNK	TEST ELEMENT CHAR FOR BLANK
348B	D0 81 81	2134	BE	FZS150(,@BR)	* AND REPEAT LOOP UNTIL RIGHT-
		2135		*	* MOST NON-BLANK CHAR IS FOUND
348E	5C 00 C7 8A	2136	FZS158 MVC	FZSCNT(,@BR),FZS155+@D1(1,@BR)	SET CHAR COUNT FOR NUMBER
		2137		*	* OF SIGNIFICANT ELEMENT CHARS
3492	F2 87 0C	2138	J	FZS170	GO SET FOR CHARACTER OUTPUT
		2139		*	
		2140		* ELEMENT IS A CHARACTER STRING SEGMENT - LEAVE ELEMENT IN STACK IN	
		2141		* THE FORM 'SEGMENT' (TRAILING BLANKS ALLOWED)	
		2142		*	
3495	1E 00 0D57 E2	2143	FZS160 ALC	I\$PARM,FZSSAJ(1,@BR)	ADJUST OUTPUT CONTROL PARAMETER
		2144		*	* FOR CHARACTER STRING SEGMENT
349A	BB E0 00	2145	SBF	I@STAT(,@XR),X'FF'-B@CCNT	SET CHAR COUNT EQUAL TO COUNT
349D	6C 00 C7 00	2146	MVC	FZSCNT(,@BR),I@STAT(1,@XR)	* FIELD IN ELEMENT STATUS BYTE.
		2147		*	
		2148		* ADJUST OUTPUT AREA POINTER FOR THE CHARACTER ELEMENT	
		2149		*	
34A1	E2 02 01	2150	FZS170 LA	@B1(,@XR),@XR	INCR POINTER PAST STATUS BYTE
		2151		*	
		2152		*****	

## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 224
				2154		*****				
				2155		* OUTPUT OPERATION INTERFACE ROUTINE				*
				2156		*****				
				2157		*				
				2158		* PAD THE CONVERTED DATA FIELD WITH BLANKS TO A FULL PRINT ZONE				
				2159		*				
34A4	7C	11	B2	2160	FZS180	MVI	FZS190+@Q(,@BR),I@LFPZ-1	SET LENGTH OF FIELD TO BE		
34A7	5F	00	B2 C7	2161		SLC	FZS190+@Q(,@BR),FZSCNT(1,@BR)	* PADDED - BYPASS PADDING		
34AB	F2	82	07	2162		JL	FZS200	* OPERATION IF LENGTH - 0		
34AE	BC	40	12	2163		MVI	I@LFPZ(,@XR),B@BLNK	PROPAGATE BLANKS TO FILL		
34B1	AC	00	11 12	2164	FZS190	MVC	I@LFPZ-1(,@XR),I@LFPZ(@VQ,@XR)	* THE FIELD TO FULL ZONE		
				2165		*				
				2166		* CONVERT THE OUTPUT PARAMETER TO AN ENTRY POINT DISPLACEMENT				
				2167		*				
34B5	34	02	0D59	2168	FZS200	ST	I\$WRK1,@XR	SAVE THE PRINT FIELD POINTER		
34B9	D2	02	E4	2169		LA	FZSCAT-1(,@BR),@XR	LOAD CONTROL ADDRESS TABLE BASF		
34BC	4C	00	C5 0D57	2170		MVC	FZS210+@OPD2(,@BR),I\$PARM(1)	SET THE TABLE DISPLACEMENT		
34C1	2C	00	0D57 00	2171	FZS210	MVC	I\$PARM,*-(1,@XR)	MOVE ENTRY PT DISP TO PARAMETER		
				2172		*				
				2173		* ESTABLISH THE DATA FIELD CHARACTER COUNT PARAMETER				
				2174		*				
34C6	3C	00	0D56	2175	FZS230	MVI	I\$PARM-1,*-	MOVE DATA FIELD COUNT TO PARAM		
				2176		*				
				2177		* ESTABLISH POSSIBLE CORE ENTRY ADDRESS FOR THE CRT IOCR				
				2178		*				
34CA	1C	01	0D5B E4	2179		MVC	I\$WRK2,FZSPDA(@CADDR,@BR)	SET BASE CRT ENTRY CORE ADDRESS		
34CF	0E	00	0D5A 043B	2180		ALC	I\$WRK2-1,\$EXFTR(1)	ADJUST CADDR FOR CORE EYTENSION		
				2181		*				
				2182		* OUTPUT THE DATA FIELD AS SPECIFIED BY CONTROL PARAMETER				
				2183		*				
34D5	C0	87	12B1	2184	FZS240	B	I\$CALL	LINK TO OUTPUT THE DATA FIELD		
34D9	3600			2185		DC	AL(@VADDR)(FZS600)	OUTPUT RIN VIRTUAL ADDRESS		
				2186		*				
				2187		* RETURN CONTROL TO THE INTERPRETER CALLING ROUTINE				
				2188		*				
34DB	C0	87	12D3	2189	FZS260	B	I\$RTRN	RETURN TO INTERPRETER		
				2190		*				
				2191		*****				

FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN							
ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00    31/05/21    PAGE 225	
			2193	*****			
			2194	*	PRINT EXECUTION ROUTINE CONSTANTS (1ST VM PAGE)		*
			2195	*****			
			2196	*			
34DF	01	34DF	2197	FZSBN1 DC	IL1'1'	BINARY INTEGER+1	
			2198	*			
34E0	80	34E0	2199	FZSNXZ DC	AL1(B@NXZR)	ZERO NORMALIZED EXPONENT	
34E1	08	34E1	2200	FZSCAJ DC	AL1(B@PRRL)	CTL PARAM ADJUST - CHAR REF	
34E2	0C	34E2	2201	FZSSAJ DC	AL1(B@PRPR+B@PRRL)	CTL PARAM ADJUST - CHAR STRING	
			2202	*			
34E3	2004	34E4	2203	FZSPDA DC	AL(@CADDR)(I\$CSXA+@INST4)	CRT IOCR CORE ENTTY ADDR BASE	
			2204	*			
			2205	*****			

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 226
			2207		*****	
			2208		* OUTPUT CONTROL PARAMETER FUNCTION ADDRESS TABLE	*
			2209		*****	
			2210		*	
			2211		* DISPLACEMENT ENTRIES IN THE FOLLOWING TABLE REFERENCE THE MATRIX	
			2212		* PRINTER OUTPUT ROUTINE (3RD VM PAGE), BUT ARE USED ALSO IN CON-	
			2213		* JUNCTION WITH THE CRT OUTPUT ROUTINE (4TH VM PAGE). THUS, 4TH PAGE	
			2214		* DISPLACEMENTS MUST BE KEPT IDENTICAL WITH 3RD PAGE DISPLACEMENTS	
			2215		* WHICH ARE REFERENCED IN THE TABLE (E.G, FOR CODE 9, FZS860-FZS810	
			2216		* MUST BE KEPT IDENTICAL TO FZS660-FZS610).	
			2217		*	
		34E5	2218	FZSCAT EQU *	CONTROL ADDR TABLE ADDRESS	
			2219	*		
34E5	00		34E5	2220	DC AL1(FZS610-FZS610)	CODE 1 - PRT ARITH, NO SPACE
34E6	18		34E6	2221	DC AL1(FZS620-FZS610)	CODE 2 - PRT ARITH, SPACE FULL
34E7	1E		34E7	2222	DC AL1(FZS630-FZS610)	CODE 3 - PRT ARITH, SPACE PACK
34E8	4D		34E8	2223	DC AL1(FZS650-FZS610)	CODE 4 - PRT ARITH, RTRN CARR
			2224	*		
34E9	59		34E9	2225	DC AL1(FZS660-FZS610)	CODE 5 - SPACE FULL
34EA	5F		34EA	2226	DC AL1(FZS670-FZS610)	CODE 6 - SPACE PACKED
34EB	73		34EB	2227	DC AL1(FZS680-FZS610)	CODE 7 - RETURN CARRIER
34EC	79		34EC	2228	DC AL1(FZS690-FZS610)	CODE 8 - RETURN CARR ON COND
			2229	*		
34ED	00		34ED	2230	DC AL1(FZS610-FZS610)	CODE 9 - PRI CHAR, NO SPACE
34EE	82		34EE	2231	DC AL1(FZS695-FZS610)	CODE 10 - PRT CHAR, SPACE FULL
34EF	00		34EF	2232	DC AL1(FZS610-FZS610)	CODE 11 - PRT CHAR, SPACE PACK
34F0	4D		34F0	2233	DC AL1(FZS650-FZS610)	CODE 12 - PRT CHAR, RTRN CARR
			2234	*		
34F1	00		34F1	2235	DC AL1(FZS610-FZS610)	CODE 13 - PRT STRING, NO SPACE
34F2	88		34F2	2236	DC AL1(FZS700-FZS610)	CODE 14 - PRT STRING, SPACE LNG
34F3	00		34F3	2237	DC AL1(FZS610-FZS610)	CODE 15 - PRT STRING, SPACE PKD
34F4	4D		34F4	2238	DC AL1(FZS650-FZS610)	CODE 16 - PRT STRING, RTRN CARR
			2239	*		
			2240		*****	
			2241		* PRINT EXECUTION ROUTINE EQUATES (1ST VM PAGE)	*
			2242		*****	
			2243		*	
		0000	2244	FZSPAL EQU 0	DISP FOR OUTPUT AREA LEFT BYTE	
			2245	*		
		34C7	2246	FZSCNT EQU FZS230+@Q	DATA CHARACTER COUNTER	
			2247	*		
			2248		*****	

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 227
					2250	*****	*****			
					2251	*	VIRTUAL MEMORY PRINT E-EXECUTION ROUTINE 2ND VM PAGE -			*
					2252	*	* ROUNDS THE ARITHMETIC VALUE IN THE RUN-TIME STACK			*
					2253	*	* CONVERTS ARITHMETIC VALUE TO E- OR F-FORMAT FOR OUTPUT			*
					2254	*				*
					2255	*	INPUT -			*
					2256	*	* RUN-TIME STACK - CONTAINS ARITHMETIC VALUE TO BE CONVERTED			*
					2257	*	* REGISTER @XR - CONTAINS CORE ADDRESS OF VALUE EXPONENT BYTE			*
					2258	*				*
					2259	*	OUTPUT -			*
					2260	*	* RUN-TIME STACK - CONTAINS CONVERTED ARITHMETIC VALUE			*
					2261	*	* REGISTER @XR - CONTAINS CORE ADDRESS OF VALUE SIGN POSITION			*
					2262	*	* I\$PARM-1 - 1 BYTE, CONTAINS VALUE CHAR COUNT (NOT INCL SIGN)			*
					2263	*****	*****			
					2264	*				
					2265	*	ESTABLISH ADDRESSABILITY FOR PRINT ROUTINE 2ND VM PAGE			
					2266	*				
					2267	*	FZSP2B VPAGE 0			
3500					2268	ORG	*,256,0	SET STARTING ADDRESS		
				3500	2269	FZSP2B EQU	*	START OF PROGRAM CODING		
3401					2270	ORG	*-255	RESET IAR TO PAGE		
3500					2271	ORG	*,256,0	* BOUNDARY ADDRESS		
				3500	2272	USING	*,@BR	SET PAGE BASE ADDRESS		
3500					2273	ORG	FZSP2B	RESET STARTING ADDRESS		
					2274	***	END OF EXPANSION ***			
					2275	*				
					2276	*	CONVERSION ENTRY - ROUND THE ARITHMETIC VALUE FOR E- OR F-FORMAT			
					2277	*				
				3500	2278	FZS300 EQU	*	CONVERSION ROUTINE ENTRY POINT		
3500 96 60 07 CC					2279	AZ	I@APRC+1(I@APRC+1,@XR),FZSDC5(1,@BR)	ROUND THE VALUE UP		
3504 F2 08 07					2280	JNOZ	FZS310	IF NO OVFLOW SKIP TO CONTINUE,		
3507 BC F1 01					2281	MVI	I@MANL(,@XR),B@DEC1	* ELSE SET MOST SIGNIFICANT		
350A 9E 00 00 CA					2282	ALC	I@DEXP(,@XR),FZS2B1(1,@BR)	* DIGIT = 1 AND INCR EXPONENT		
					2283	*				
					2284	*	TEST MAGNITUDE OF VALUE FOR OUTPUT IN E- OR F-FORMAT			
					2285	*				
350E BD 80 00					2286	FZS310 CLI	I@DEXP(,@XR),B@NXZR	IF VALUE LESS THAN 1E-1, OR		
3511 D0 82 4D					2287	BL	FZS400(,@BR)	* GREATER THAN OR EQUAL TO		
3514 BD 86 00					2288	CLI	I@DEXP(,@XR),B@NXZR+I@APRC	* 1E+6 (1E+11 FOR LONG PREC),		
3517 D0 84 4D					2289	BH	FZS400(,@BR)	* GO CONVERT VALUE TO E-FORMAT		
					2291	*****	*****			
					2292	*	F-FORMAT OUTPUT CONVERSION ROUTINE			*
					2293	*****	*****			
					2294	*				
					2295	*	SHIFT FRACTIONAL-COMPONENT RIGHT TO INSERT DECIMAL POINT			
					2296	*				
351A 7C 85 25					2297	FZS320 MVI	FZS330+@Q(,@BR),B@NXZR+I@APRC-1	ESTABLISH LENGTH CODE FOR		
351D 6F 00 25 00					2298	SLC	FZS330+@Q(,@BR),I@DEXP(1,@XR)	* FRACTIONAL COMPONENT		
3521 F2 82 04					2299	JL	FZS340	BRANCH IF NO FRACTION		
3524 AC 00 07 06					2300	FZS330 MVC	I@APRC+1(,@XR),I@APRC(@VQ,@XR)	SHIFT FRACTION RIGHT BY 1		
					2301	*				
					2302	*	ESTABLISH F-FORMAT DECIMAL POINT - VALUE IS LEFT IN STACK IN FORM			
					2303	*	'S.123456', S123.456', OR 'S123456.' WHERE 'S' IS THE SIGN POSITION			
					2304	*				
3528 6C 00 36 00					2305	FZS340 MVC	FZS350+@D1(,@BR),I@DEXP(1,@XR)	CALCULATE DISPLACEMENT		



ERR LOC		OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21		PAGE 228
352C	5E 00 36	CA	2306		ALC	FZS350+@D1(,@BR),FZS2B1(1,@BR) * FOR THE DECIMAL POINT			
3530	5F 00 36	CD	2307		SLC	FZS350+@D1(,@BR),FZS2XZ(1,@BR) * IN F-FORMAT FIELD			
3534	BC 4B 00		2308	FZS350	MVI	*-*(,@XR),B@DPNT INSERT THE DECIMAL POINT			
			2309		*				
			2310		*	TRUNCATE INSIGNIFICANT ZEROS FROM THE ROUNDED VALUE			
			2311		*				
3537	7C 08 40		2312	FZS360	MVI	FZS380+@D1(,@BR),I@APRC+2 SET DISP FOR BYTE AFTER VALUE			
353A	5F 00 40	CA	2313	FZS370	SLC	FZS380+@D1(,@BR),FZS2B1(1,@BR) DECR VALUE CHAR POINTER			
353E	BD F0 00		2314	FZS380	CLI	*-*(,@XR),B@DEC0 TEST VALUE CHARACTER FOR ZERO			
3541	D0 81 3A		2315		BE	FZS370(,@BR) * AND REPEAT UNTIL NON-ZERO			
			2316		*				
			2317		*	SET COUNT PARAMETER AND RETURN TO CALLING PAGE			
			2318		*				
3544	1C 00 0D56	40	2319	FZS390	MVC	I\$PARM-1,FZS380+@D1(1,@BR) MOVE DATA CHARACTER COUNT			
			2320		*	* TO THE OUTPUT PARAMETER			
3549	C0 87 12D3		2321		B	I\$RTRN RETURN TO CALLING PAGE			
			2322		*				
			2323			*****			

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 229
				2325		*****				
				2326	*	E-FORMAT OUTPUT CONVERSION ROUTINE				*
				2327		*****				
				2328	*					
				2329	*	SHIFT MANTISSA (EXCEPT MOST SIGNIFICANT DIGIT) RIGHT TO INSERT				
				2330	*	DECIMAL POINT - ESTABLISH E-FORMAT DECIMAL POINT, LEAVING VALUE				
				2331	*	IN STACK IN FORM 'S1.23496' WHERE 'S' IS THE SIGN POSITION				
				2332	*					
354D	AC	04	07	06	2333	FZS400 MVC	I@APRC+1(,@XR),I@APRC(I@APRC-1,@XR) SHIFT MANTISSA RIGHT			
3551	BC	4B	02		2334	MVI	FZSPAL+2(,@XR),B@DPNT INSERT E-FORMAT DECIMAL POINT			
3554	9F	00	00	CA	2335	SLC	I@DEXP(,@XR),FZS2B1(1,@BR) ADJUST EXPONENT TO COMPENSATE			
				2336	*					
				2337	*	TRUNCATE INSIGNIFICANT ZEROS FROM ROUNDED VALUE - KEEP AT LEAST ONE				
				2338	*	DIGIT TO RIGHT OF DECIMAL POINT				
				2339	*					
3558	BB	F0	03		2340	FZS410 SBF	FZSPAL+3(,@XR),B@ZPOS FLAG DIGIT AFTER DECIMAL POINT			
355B	7C	08	64		2341	MVI	FZS430+@D1(,@BR),I@APRC+2 SET DISP FOR BYTE AFTER VALUE			
355E	5F	00	64	CA	2342	FZS420 SLC	FZS430+@D1(,@BR),FZS2B1(1,@BR) DECR VALUE CHAR POINTER			
3562	BD	F0	00		2343	FZS430 CLI	*-(,@XR),B@DEC0 TEST VALUE CHARACTER FOR ZERO			
3565	D0	81	5E		2344	BE	FZS420(,@BR) * AND REPEAT UNTIL NON-ZERO			
3568	BA	F0	03		2345	FZS435 SBN	FZSPAL+3(,@XR),B@ZPOS RESTORE DIGIT AFTER DEC POINT			
				2346	*					
				2347	*	SET COUNT PARAMETER FOR FORMATTED MANTISSA PLUS 4 BYTE EXPONENT				
				2348	*					
356B	3C	04	0D56		2349	FZS440 MVI	I\$PARM-1,FZSLXB SET DATA CHAR CNT FOR EXPONENT			
356F	1E	00	0D56	64	2350	ALC	I\$PARM-1,FZS430+@D1(1,@BR) INCR DATA CHAR COUNT FOR VALUE			
				2351	*					
				2352	*	INITIALIZE OUTPUT FORM OF EXPONENT - TEST FOR EXPONENT SIGN				
				2353	*					
3574	5C	03	D6	D1	2354	FZS450 MVC	FZSXWK(,@BR),FZSEXB(FZSLXB,@BR) MOVE EXPONENT IMAGE TO			
				2355	*		* EXPONENT WORK AREA			
3578	6C	00	D2	00	2356	MVC	FZS2BX(,@BR),I@DEXP(1,@XR) DETERMINE BINARY MAGNITUDE			
357C	5F	00	D2	CD	2357	SLC	FZS2BX(,@BR),FZS2XZ(1,@BR) * ASSUMING POSITIVE EXPONENT			
3580	F2	81	29		2358	JE	FZS480 BRANCH IF EXPONENT IS ZERO			
3583	F2	84	0A		2359	JH	FZS470 BRANCH IF EXPONENT IF POSITIVE			
				2360	*					
				2361	*	NEGATIVE EXPONENT - MODIFY SIGN AND RECOMPUTE BINARY EXPONENT				
				2362	*					
3586	7C	60	D4		2363	FZS460 MVI	FZSXWK-FZSLXM(,@BR),B@MINS MAKE EXPONENT SIGN NEGATIVE			
3589	7C	80	D2		2364	MVI	FZS2BX(,@BR),B@NXZR DETERMINE BINARY MAGNITUDE			
358C	6F	00	D2	00	2365	SLC	FZS2BX(,@BR),I@DEXP(1,@XR) * FOR NEGATIVE EXPONENT			
				2366	*					
				2367	*	CONVERT BINARY EXPONENT MAGNITUDE TO ZONED DECIMAL				
				2368	*					
3590	54	10	D8	CB	2369	FZS470 ZAZ	FZSDAC(FZSLXM,@BR),FZSDC1(1,@BR) SET DEC ACCUMULATOR = 1			
3594	7C	01	98		2370	MVI	FZS472+@Q(,@BR),@B1 SET BINARY MASK FOR 2**0 BIT			
3597	78	00	D2		2371	FZS472 TBN	FZS2BX(,@BR),*- TEST BINARY EXP MAGNITUDE BIT			
359A	F2	90	04		2372	JF	FZS474 * AND BRANCH IF BIT IS ZERO			
359D	56	01	D6	D8	2373	AZ	FZSXWK(FZSLXM,@BR),FZSDAC(FZSLXM,@BR) INCR DECIMAL EXP			
35A1	5E	00	98	98	2374	FZS474 ALC	FZS472+@Q(,@BR),FZS472+@Q(1,@BR) SHIFT BINARY MASK LEFT			
35A5	56	01	D8	D8	2375	AZ	FZSDAC(FZSLXM,@BR),FZSDAC(FZSLXM,@BR) DOUBLE DEC ACCUM			
35A9	D0	08	97		2376	BNOZ	FZS472(,@BR) REPEAT LOOP UNTIL ACCUM > 644			
				2377	*					
				2378	*	TEST FOR AND DELETE ANY INSIGNIFICANT ZERO IN THE DECIMAL EXPONENT				
				2379	*					
35AC	7D	F0	D5		2380	FZS480 CLI	FZSXWK-1(,@BR),B@DEC0 TEST FOR EXPONENT LEFTMOST ZERO			

FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN									
ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15,	MOD 00 31/05/21 PAGE 230
	35AF	F2	01 09		2381	JNE	FZS490	BRANCH IF NO INSIGNIFICANT ZERO	
	35B2	5C	00 D5 D6		2382	MVC	FZSXWK-1(,@BR),FZSXWK(1,@BR)	SHIFT SIGNIFICANT DIGIT	
	35B6	1F	00 0D56 CA		2383	SLC	I\$PARM-1,FZS2B1(1,@BR)	DECREMENT DATA CHARACTER COUNT	
					2384	*			
					2385	*	MOVE OUTPUT FORM OF EXPONENT TO THE DATA PRINT FIELD		
					2386	*			
	35BB	7C	04 C4		2387	FZS490 MVI	FZS500+@D1(,@BR),FZSLXB	SET DIP TO ESTABLISH	
	35BE	5E	00 C4 64		2388	ALC	FZS500+@D1(,@BR),FZS430+@D1(1,@BR)	* EXPONENT POSITION	
	35C2	9C	03 00 D6		2389	FZS500 MVC	*-*(,@XR),FZSXWK(FZSLXB,@BR)	MOVE EXPONENT TO PRINT FIELD	
					2390	*			
					2391	*	RETURN CONTROL TO THE CALLING PAGE		
					2392	*			
	35C6	C0	87 12D3		2393	FZS510 B	I\$RTRN	RETURN TO CALLING PAGE	
					2394	*			
					2395	*****			

FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN									
ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15,	MOD 00 31/05/21 PAGE 230
	35AF	F2	01 09		2381	JNE	FZS490	BRANCH IF NO INSIGNIFICANT ZERO	
	35B2	5C	00 D5 D6		2382	MVC	FZSXWK-1(,@BR),FZSXWK(1,@BR)	SHIFT SIGNIFICANT DIGIT	
	35B6	1F	00 0D56 CA		2383	SLC	I\$PARM-1,FZS2B1(1,@BR)	DECREMENT DATA CHARACTER COUNT	
					2384	*			
					2385	*	MOVE OUTPUT FORM OF EXPONENT TO THE DATA PRINT FIELD		
					2386	*			
	35BB	7C	04 C4		2387	FZS490 MVI	FZS500+@D1(,@BR),FZSLXB	SET DIP TO ESTABLISH	
	35BE	5E	00 C4 64		2388	ALC	FZS500+@D1(,@BR),FZS430+@D1(1,@BR)	* EXPONENT POSITION	
	35C2	9C	03 00 D6		2389	FZS500 MVC	*-*(,@XR),FZSXWK(FZSLXB,@BR)	MOVE EXPONENT TO PRINT FIELD	
					2390	*			
					2391	*	RETURN CONTROL TO THE CALLING PAGE		
					2392	*			
	35C6	C0	87 12D3		2393	FZS510 B	I\$RTRN	RETURN TO CALLING PAGE	
					2394	*			
					2395	*****			

FZSPRT	- S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN		
ERR LOC	OBJECT CODE	ADDR STMT SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 230
35AF F2 01 09		2381 JNE FZS490 BRANCH IF NO INSIGNIFICANT ZERO	
35B2 5C 00 D5 D6		2382 MVC FZSXWK-1(,@BR),FZSXWK(1,@BR) SHIFT SIGNIFICANT DIGIT	
35B6 1F 00 0D56 CA		2383 SLCL I\$PARM-1,FZS2B1(1,@BR) DECREMENT DATA CHARACTER COUNT	
		2384 *	
		2385 * MOVE OUTPUT FORM OF EXPONENT TO THE DATA PRINT FIELD	
		2386 *	
35BB 7C 04 C4		2387 FZS490 MVI FZS500+@D1(,@BR),FZSLXB SET DIP TO ESTABLISH	
35BE 5E 00 C4 64		2388 ALC FZS500+@D1(,@BR),FZS430+@D1(1,@BR) * EXPONENT POSITION	
35C2 9C 03 00 D6		2389 FZS500 MVCL *-*(,@XR),FZSXWK(FZSLXB,@BR) MOVE EXPONENT TO PRINT FIELD	
		2390 *	
		2391 * RETURN CONTROL TO THE CALLING PAGE	
		2392 *	
35C6 C0 87 12D3		2393 FZS510 B I\$RTRN RETURN TO CALLING PAGE	
		2394 *	
		2395 *****	

[illegible][illegible][illegible]

FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN									
ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15,	MOD 00 31/05/21 PAGE 230
	35AF	F2	01 09		2381	JNE	FZS490	BRANCH IF NO INSIGNIFICANT ZERO	
	35B2	5C	00 D5 D6		2382	MVC	FZSXWK-1(,@BR),FZSXWK(1,@BR)	SHIFT SIGNIFICANT DIGIT	
	35B6	1F	00 0D56 CA		2383	SLC	I\$PARM-1,FZS2B1(1,@BR)	DECREMENT DATA CHARACTER COUNT	
					2384	*			
					2385	*	MOVE OUTPUT FORM OF EXPONENT TO THE DATA PRINT FIELD		
					2386	*			
	35BB	7C	04 C4		2387	FZS490 MVI	FZS500+@D1(,@BR),FZSLXB	SET DIP TO ESTABLISH	
	35BE	5E	00 C4 64		2388	ALC	FZS500+@D1(,@BR),FZS430+@D1(1,@BR)	* EXPONENT POSITION	
	35C2	9C	03 00 D6		2389	FZS500 MVC	*-*(,@XR),FZSXWK(FZSLXB,@BR)	MOVE EXPONENT TO PRINT FIELD	
					2390	*			
					2391	*	RETURN CONTROL TO THE CALLING PAGE		
					2392	*			
	35C6	C0	87 12D3		2393	FZS510 B	I\$RTRN	RETURN TO CALLING PAGE	
					2394	*			
					2395	*****			

[illegible]

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 231
			2397	*****	*****	
			2398	* PRINT EXECUTION ROUTINE CONSTANTS (2ND VM PAGE)	*	
			2399	*****	*****	
			2400	*		
35CA	01	35CA	2401	FZS2B1 DC	IL1'1' BINARY INTEGER +1	
35CB	F1	35CB	2402	FZSDC1 DC	DL1'1' DECIMAL INTEGER +1	
35CC	F5	35CC	2403	FZSDC5 DC	DL1'5' DECIMAL INTEGER +5	
			2404	*		
35CD	80	35CD	2405	FZS2XZ DC	AL1(B@NXZR) ZERO NORMALIZED EXPONENT	
			2406	*		
		0004	2407	FZSLXB EQU	4 LENGTH OF EXPONENT IMAGE	
35CE	C54EF0F0	35D1	2408	FZSEXB DC	CL(FZSLXB)'E+00' EXPONENT IMAGE FOR OUTPUT	
			2409	*		
			2410	*****	*****	
			2411	* PRINT EXECUTION ROUTINE WORK AREAS (2ND VM PAGE)	*	
			2412	*****	*****	
			2413	*		
35D2		35D2	2414	FZS2BX DS	CL1 BINARY EXPONENT MAGNITUDE	
35D3		35D6	2415	FZSXWK DS	CL(FZSLXB) EXPONENT CONSTRUCT AREA	
			2416	*		
		0002	2417	FZSLXM EQU	2 LENGTH OF DECMAL EXP MAGNITUDE	
35D7		35D8	2418	FZSDAC DS	CL(FZSLXM) B TO D DECIMAL ACCUMULATOR	
			2419	*		
			2420	*****	*****	

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 232
				2422	*****	
				2423	* VIRTUAL MEMORY PRINT EXECUTION ROUTINE (3RD VM PAGE)	*
				2424	* * OUTPUTS FORMATTED DATA ELEMENT TO MATRIX PRINTER	*
				2425	* * CONTROLS PRINTER CARRIER DEPENDING ON SPECIFIED CONTROL CODE	*
				2426	* INPUT -	*
				2427	* * RUN-TIME STACK - CONTAINS FORMATTED ELEMENT, IF PRESENT	*
				2428	* * I\$PARM - 1 BYTE, CONTAINS CONTROL CODE BRANCH DISPLACEMENT	*
				2429	* * I\$PARM-1 - 1 BYTE, CONTAINS FORMATTED ELEMENT CHARACTER COUNT	*
				2430	* * I\$WRK1 - 2 BYTES, CONTAINS CORE ADDR OF PRINT AREA LEFT BYTE	*
				2431	* * I\$WRK2 - 2 BYTES, CONTAINS VALUE FOR \$PRDEV 'CRT ONLY' COND	*
				2432	* * I\$SLLC - 1 BYTE, CONTAINS OUTPUT ELEMENT LENGTH CODE (LNG - 1)	*
				2433	*	*
				2434	* OUTPUT -	*
				2435	* * PRINTED ELEMENT AND/OR CARRIER CONTROL ON MATRIX PRINTER	*
				2436	*****	
				2437	*	
				2438	* ESTABLISH ADDRESSABILITY FOR PRINT ROUTINE (3RD VM PAGE)	
				2439	*	
				2440	*FZSP3B VPAGE 0	
3600				2441	ORG *,256,0	SET STARTING ADDRESS
	3600			2442	FZSP3B EQU *	START OF PROGRAM CODING
3501				2443	ORG *-255	RESET IAR TO PAGE
3600				2444	ORG *,256,0	* BOUNDARY ADDRESS
	3600			2445	USING *,@BR	SET PAGE BASE ADDRESS
3600				2446	ORG FZSP3B	RESET STARTING ADDRESS
				2447	*** END OF EXPANSION ***	
				2448	*	
				2449	* PAGE ENTRY - TEST FOR MATRIX PRINTER ACTIVE ON SYSTEM	
				2450	*	
3600 0D 01 044B 0D5B				2451	FZS600 CLC \$PRDEV,I\$WRK2(@CADDR)	IF PRINTER NOT A SYSTEM PRINT ?
3606 F2 02 BF				2452	JNL FZS740	* DEVICE, GO OUTPUT TO THE CRT
				2453	*	
				2454	* INITIALIZE FOR OUTPUT TO THE MATRIX PRINTER	
				2455	*	
3609 4C 00 6A 03C0				2456	MVC FZS3RM(,@BR),\$RMRGN(1)	SET MP RIGHT MARGIN PARAMETER
				2457	*	
				2458	* INITIALIZE THE ELEMENT PRINT PARAMETER LIST	
				2459	*	
360E 7C 40 F2				2460	MVI FZS3PF(,@BR),@PRINT	SET FUNCTION FOR PRINT ONLY
3611 4C 00 F3 0D56				2461	MVC FZS3PC(,@BR),I\$PARM-1(1)	SET COUNT = ELEMENT CHAR COUNT
3616 4C 01 F5 0D59				2462	MVC FZS3PA(,@BR),I\$WRK1(@CADDR)	SET PRINT AREA CORE ADDRESS
				2463	*	
				2464	* TEST FOR AN ARITHMETIC ELEMENT - RETURN CARRIER IF ARITHMETIC	
				2465	* ELEMENT LENGTH EXCEEDS OUTPUT LINE MARGIN	
				2466	*	
361B 5C 00 DB F3				2467	MVC FZS3CC(,@BR),FZS3PC(1,@BR)	SET PARAM = ELEMENT CHAR CNT
361F 3D 12 0BA1				2468	CLI I\$SLLC,I@LCRV-1	IF CURR ELEMENT IS ARITHMETIC ?
3623 D0 01 D2				2469	BNE FZS760(,@BR)	* LINK TO RETURN CARR ON COND
				2470	*	
				2471	* BRANCH TO APPROPRIATE ROUTINE DEPENDING ON CONTROL CODE	
				2472	*	
3626 4C 00 2D 0D57				2473	MVC FZS605+@D1(,@BR),I\$PARM(1)	MOVE CONTROL DISP TO JUMP INST
362B F2 87 00				2474	FZS605 J *-*	GO EXECUTE CONTROL CODE ROUTINE
				2475	*	
				2476	*****	

## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 233
				2478			*****			
				2479	*		OUTPUT ROUTINE FOR PRINT CONTROL CODES 1, 9, 11, 13, 15			*
				2480			*****			
				2481	*					
				2482	*		PRINT THE FORMATTED ELEMENT ONLY (WHEN SIGNIFICANT)			
				2483	*					
362E	7D	00	F3	2484	FZS610	CLI	FZS3PC(,@BR),@ZERO		IF ELEMENT CHAR COUNT NOT ZERO	
3631	F2	81	9A	2485		JE	FZS750		EXIT ROUTINE W/O PRINTING	1-5
3634	1C	01	144A	2486		MVC	I\$VADR,FZSPCH(@VADDR,@BR)		VM PATCH PAGE ENTRY ADDR	1-5
3639	C0	87	1358	2487		B	I\$CVAD		LOAD PATCH PAGE	1-5
363D	4C	01	45 144C	2488		MVC	FZS615+@OP1(@CADDR,@BR),I\$CADR		MOVE CADDR TO BRANCH	1-5
3642	C0	87	0000	2489	FZS615	B	*-*		BRANCH TO PATCH PAGE	1-5
				2490	*					
				2491			*****			
				2492	*		OUTPUT ROUTINE FOR PRINT CONTROL CODE 2			*
				2493			*****			
				2494	*					
				2495	*		ESTABLISH FULL PRINT ZONE OUTPUT FORMAT (ARITHMETIC ELEMENT)			
				2496	*					
3646	7C	12	DB	2497	FZS620	MVI	FZS3CC(,@BR),I@LFPZ		SET PARAM - FULL PRINT ZONE	
3649	F2	87	18	2498		J	FZS636		BRANCH TO TEST LINE CAPACITY	
				2499	*					
				2500			*****			

## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 234
		2502		*****	
		2503		* OUTPUT ROUTINE FOR PRINT CONTROL CODE 3	*
		2504		*****	
		2505		*	
		2506		* ESTABLISH PACKED PRINT ZONE OUTPUT FORMAT (ARITHMETIC ELEMENT) -	
		2507		* THIS ZONE WILL BE 6, 9, 12, 15, OR 18 CHARACTERS LONG DEPENDING ON	
		2508		* THE LENGTH OF THE ARITHMETIC ELEMENT TO BE PRINTED.	
		2509		*	
364C	7C 04 DB	2510	FZS630 MVI	FZS3CC(,@BR),2*I@LPPZ-2	SET LENGTH ACCUM TO MINIMUM
		2511	*		* ELEMENT LENGTH LIMIT (4)
364F	5D 00 F3 DB	2512	FZS632 CLC	FZS3PC(,@BR),FZS3CC(1,@BR)	IF ELEMENT LENGTH WITHIN LIMIT
3653	F2 04 0A	2513	JNH	FZS634	* BRANCH TO EXIT THIS LOOP
3656	5E 00 DB F1	2514	ALC	FZS3CC(,@BR),FZS3PZ(1,@BR)	ADD PACKED ZONE INCR TO ACCUM
365A	7D 10 DB	2515	FZS633 CLI	FZS3CC(,@BR),I@LFPZ-2	IF LENGTH ACCUM NOT MAXIMUM
365D	D0 82 4F	2516	BL	FZS632(,@BR)	* GO REPEAT ELEMENT LENGTH TEST
		2517	*		
3660	5E 00 DB F0	2518	FZS634 ALC	FZS3CC(,@BR),FZS3B2(1,@BR)	ADJUST ACCLM TO MAKE PACKED
		2519	*		* PRINT ZONE FIELD LENGTH
		2520	*		
		2521	*	TEST LINE CAPACITY TO CONTAIN CURRENT PRINT ZONE FIELD - WHEN RIGHT	
		2522	*	MARGIN IS EXCEEDED, LINE HAS CAPACITY FOR THE DATA ELEMENT BUT NOT	
		2523	*	FOR THE ENTIRE PRINT ZONE ... IN THIS CASE, PRINT ELEMENT ONLY AND	
		2524	*	RETURN THE CARRIER	
		2525	*		
3664	4E 00 DB 03C2	2526	FZS636 ALC	FZS3CC(,@BR),\$PRPOS(1)	ADD PRINT ZONE LNG TO CURRENT
3669	7D 00 DB	2527	FZS638 CLI	FZS3CC(,@BR),*-*	* CARRIER POSITION - BRANCH
366C	F2 84 12	2528	JH	FZS655	* IF RIGHT MARGIN IS EXCEEDED
		2529	*		
		2530	*	LINE HAS CAPACITY FOR ENTIRE PRINT ZONE - PRINT ELEMENT AND SPACE	
		2531	*	TO THE SPECIFIED ZONE POSITION	
		2532	*		
366F	4F 00 DB 03C2	2533	FZS640 SLC	FZS3CC(,@BR),\$PRPOS(1)	RESTORE CURRENT PRINT ZONE LNG
3674	5C 00 F3 DB	2534	MVC	FZS3PC(,@BR),FZS3CC(1,@BR)	SET COUNT - CAR PRT ZONE LNG
3678	F2 87 3E	2535	J	FZS710	GO PRINT ELEMENT AND SPACE CARR
		2536	*		
		2537		*****	



## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 235
		2539		*****	
		2540		* OUTPUT ROUTINE FOR PRINT CONTROL CODES 4, 12, 16	*
		2541		*****	
		2542		*	
		2543		* TEST ELEMENT SIGNIFICANCE - RETURN CARRIER ONLY WHEN NOT SIGNIFICANT	
		2544		*	
367B	7D 00 F3	2545	FZS650	CLI FZS3PC(, @BR), @ZERO	ELEMENT CHAR COUNT IS ZERO ?
367E	F2 81 20	2546		JE FZS680	* GO RETURN THE CARRIER ONLY
		2547		*	
		2548		* ELEMENT IS SIGNIFICANT - PRINT ELEMENT AND RETURN CARRIER	
		2549		*	
3681	7C C0 F2	2550	FZS655	MVI FZS3PF(, @BR), @PRETR	SET PRINT & CARR RETURN FUNC
3684	F2 87 32	2551		J FZS710	GO PRINT ELEMENT AND RTRN CARR
		2553		*****	
		2554		* OUTPUT ROUTINE FOR PRINT CONTROL CODE 5	*
		2555		*****	
		2556		*	
		2557		* ESTABLISH FULL PRINT ZONE SPACING ONLY	
		2558		*	
3687	7C 12 F3	2559	FZS660	MVI FZS3PC(, @BR), I@LFPZ	SET COUNT FOR FULL PRINT ZONE
368A	F2 87 03	2560		J FZS675	BRANCH TO EXECUTE SPACING
		2562		*****	
		2563		* OUTPUT ROUTINE FOR PRINT CONTROL CODE 6	*
		2564		*****	
		2565		*	
		2566		* ESTABLISH PACKED PRINT ZONE INCREMENT SPACING ONLY	
		2567		*	
368D	7C 03 F3	2568	FZS670	MVI FZS3PC(, @BR), I@LPPZ	SET COUNT FOR PACKED ZONE INCR
		2569		*	
		2570		* PRINT CURRENT ZONE SPACE, OR RETURN CARRIER IF END OF LINE IS HIT	
		2571		*	
3690	5C 00 DB F3	2572	FZS675	MVC FZS3CC(, @BR), FZS3PC(1, @BR)	SET PARAM FOR CURRENT ZONE LNG
3694	D0 87 D2	2573		B FZS760(, @BR)	LINK TO RETURN CARRIER ON COND
3697	5D 00 DB 6A	2574		CLC FZS3CC(, @BR), FZS3RM(1, @BR)	IF CARRIER WAS NOT RETURNED
369B	F2 04 1B	2575		JNH FZS710	* GO PRINT CURRENT ZONE SPACE,
369E	F2 87 2D	2576		J FZS750	* ELSE EXIT RTN W/0 PRINTING
		2577		*	
		2578		*****	

## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	31/05/21	PAGE 236
		2580		*****			
		2581		* OUTPUT ROUTINE FOR PRINT CONTROL CODE 7			*
		2582		*****			
		2583		*			
		2584		* ESTABLISH CARRIER RETURN ONLY			
		2585		*			
36A1	D2 02 F6	2586	FZS680	LA FZS3CR(,@BR),@XR		LOAD CARRIER RETURN PPL CADDR	
36A4	F2 87 15	2587		J FZS720		GO EXECUTE CARRIER RETURN	
		2589		*****			
		2590		* OUTPUT ROUTINE FOR PRINT CONTROL CODE 8			*
		2591		*****			
		2592		*			
		2593		* RETURN CARRIER IF FULL PRINT ZONE EXCEEDS LINE CAPACITY			
		2594		*			
36A7	7C 12 DB	2595	FZS690	MVI FZS3CC(,@BR),I@LFPZ		SET PARAM FOR PRINT ZONE	
36AA	D0 87 D2	2596		B FZS760(,@BR)		LINK TO RETURN CARRIER ON COND	
36AD	F2 87 0F	2597		J FZS730		GO TEST FOR CRT ACTIVE ON SYSTEM	
		2599		*****			
		2600		* OUTPUT ROUTINE FOR PRINT CONTROL CODE 10			*
		2601		*****			
		2602		*			
		2603		* RETURN CARRIER IF FULL PRINT ZONE EXCEEDS LINE CAPACITY			
		2604		*			
36B0	7C 12 DB	2605	FZS695	MVI FZS3CC(,@BR),I@LFPZ		SET PARAM FOR FULL PRINT ZONE	
36B3	D0 87 D2	2606		B FZS760(,@BR)		LINK TO RETURN CARRIER ON COND	
		2607		*			
		2608		*****			

## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 237
		2610		*****	
		2611		* OUTPUT ROUTINE FOR PRINT CONTROL CONTROL CODE 14	
		2612		*****	
		2613		*	
		2614		* ESTABLISH FULL PRINT ZONE OUTPUT FORMAT (CHARACTER ELEMENT)	
		2615		*	
36B6	7C 12 F3	2616	FZS700 MVI	FZS3PC(,@BR),I@LFPZ SET COUNT FOR ZONE	
		2617		*	
		2618		* EXECUTE ELEMENT OUTPUT TO THE MATRIX PRINTER	
		2619		*	
36B9	D2 02 F2	2620	FZS710 LA	FZS3PL(,@BR),@XR LOAD DATA OLTOLT CORE ADOR	
36BC	D0 87 E3	2621	FZS720 B	FZS780(,@BR) LINK TO EXECUTE PRINTER OUTPUT	
		2622		*	
		2623		* TEST FOR THE CRT ACTIVE AS A SISTEM PRINT DEVICE	
		2624		*	
36BF	0D 00 044A 0D5A	2625	FZS730 CLC	\$PRDEV-1,I\$WRK2-1(1) IF CRT IS NOT A SYSTEM PRINT	
36C5	F2 82 06	2626	JL	FZS750 * DEVICE, GO EXIT THIS ROUTINE	
		2627		*	
		2628		* CRT ACTIVE - SET UP AND OUTPUT TO CRT USINS CRT LINE WIDTH	
		2629		*	
36C8	C0 87 12B1	2630	FZS740 B	I\$CALL LINK TO EXECUTE PRINT ON CRT	
36CC	3700	2631	DC	AL(@VADDR)(FZS800) PRINT CRT RTN VIRTUAL ADDRESS	
		2632		*	
		2633		* RETURN TO PTINT ROUTINE 1ST VM PAGE	
		2634		*	
36CE	C0 87 12D3	2635	FZS750 B	I\$RTRN RETURN TO 1ST PRINT RTN PAGE	
		2636		*	
		2637		*****	

## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 238
		2639		*****	
		2640	*	PRINTER CARRIER RETURN ROUTINE -	*
		2641	*	* RETURNS PRINTER CARRIER WHEN SPECIFIED LENGTH PARAMETER	*
		2642	*	(FZS3CC) EXCEEDS THE CURRENT PRINT LINE CAPACITY.	*
		2643		*****	
		2644	*		
36D2	74 08 EF	2645	FZS760 ST	FZS790+@OP1(,@BR),@ARR STORE RETURN BRANCH ADDRESS	
		2646	*		
		2647	*	TEST LINE CAPACITY TO CONTAIN CURRENT PRINT REGION LENGTH	
		2648	*		
36D5	4E 00 DB 03C2	2649	ALC	FZS3CC(,@BR),\$PRPOS(1) ADD PRINT REGION LENGTH TO CURR	
36DA	7D 00 6A	2650	FZS770 CLI	FZS3RM(,@BR),*-*	* CARRIER POSITION - BRANCH IF
36DD	F2 02 0C	2651	JNL	FZS790	* RIGHT MARGIN NOT EXCEEDED
		2652	*		
		2653	*	RIGHT MARGIN EXCEEDED - RETURN MATRIX PRINTER CARRIER	
		2654	*		
36E0	D2 02 F6	2655	LA	FZS3CR(,@BR),@XR LOAD CARRIER RETURN PPL CADDR	
		2657		*****	
		2658	*	PRINTER OUTPUT INTERFACE -	*
		2659	*	* EXECUTES MATRIX PRINTER OUTPUT AS SPECIFIED IN PRINT PARAM-	*
		2660	*	ETER LIST REFERENCED BY REGISTER @XR.	*
		2661		*****	
36E3	74 08 EF	2662	FZS780 ST	FZS790+@OP1(,@BR),@ARR STORE RETURN BRANCH ADDRESS	
36E6	C0 87 12B1	2663	B	I\$CALL LINK TO EXECUTE PRINTER IOCR	
36EA	2800	36EB 2664	DC	AL(@VADDR)(V\$SPRT) MATRIX PRINTER IOCR VADDR	
		2665	*		
		2666	*	RETURN TO CALLING ROUTINE	
		2667	*		
36EC	C0 87 0000	2668	FZS790 B	*-* RETURN BRANCH	
		2669		*****	

## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 239
		2671		*****	
		2672		* PRINT EXECUTION ROUTINE CONSTANTS (3RD VM PAGE)	*
		2673		*****	
		2674		*	
36F0 02		36F0 2675	FZS3B2 DC	IL1'2'	BINARY INTEGER +2
		2676		*	
36F1 03		36F1 2677	FZS3PZ DC	AL1(I@LPPZ)	LENGTH OF PACKED ZONE INCR
		2679		*****	
		2680		* PRINT EXECUTION ROUTINE WORK AREAS (3RD VM PAGE)	*
		2681		*****	
		366A 2682	FZS3RM EQU	FZS638+@Q	MATRIX PRINTER RIGHT MARGIN
		36DB 2683	FZS3CC EQU	FZS770+@Q	PRINT AREA CHARACTER COUNT
		2684		*	
		2685		*FZS3PL PPL	
		36F2 2686	FZS3PL EQU	*	PPL ADDRESS
36F2 00		36F2 2687		DC AL1(*-*)	FUNCTION REQUESTED
36F3 00		36F3 2688		DC AL1(*-*)	PRINT COUNT
36F4 0000		36F5 2689		DC AL2(*-*)	DATA ADDRESS
		2690		*** END OF EXPANSION ***	
		2691		*	
		36F2 2692	FZS3PF EQU	FZS3PL+@PCTRL	PRINT FUNCTION PARAMETER
		36F3 2693	FZS3PC EQU	FZS3PL+@PRCNT	PRINT AREA COUNT PARAMETER
		36F5 2694	FZS3PA EQU	FZS3PL+@PDATA	PRINT AREA COUNT PARAMETER
		2695		*	
		2696		*FZS3CR PPL FUNC-@RETRN,CNT-@RTRNC	
		36F6 2697	FZS3CR EQU	*	PPL ADDRESS
36F6 80		36F6 2698		DC AL1(@RETRN)	FUNCTION REQUESTED
36F7 80		36F7 2699		DC AL1(@RTRNC)	PRINT COUNT
36F8 0000		36F9 2700		DC AL2(*-*)	DATA ADDRESS
		2701		*** END OF EXPANSION ***	
		2702		*	
36FA 5359		36FB 2703	FZSPCH DC	AL2(V\$PCH2+FZS633-@Q-FZSP3B)	PATCH PAGE ENTRY ADDR 1-3
		2704		*****	

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 240
		2706		*****	
		2707	*	VIRTUAL MEMORY PRINT EXECUTION ROUTINE 4TH VM PAGE	*
		2708	*	* OUTPUTS FORMATTED DATA ELEMENT TO CRT DISPLAY UNIT	*
		2709	*	* CONTROLS CRT CURSOR DEPENDING ON SPECIFIED CONTROL CODE	*
		2710	*		*
		2711	*	INPUT -	*
		2712	*	* RUN-TIME STACK - CONTAINS FORMATTED ELEMENT, IF PRESENT	*
		2713	*	* I\$PARM - 1 BYTE, CONTAINS CONTROL CODE BRANCH DISPLACEMENT	*
		2714	*	* I\$PARM-1 - 1 BYTE, CONTAINS FORMATTED ELEMENT CHARACTER COUNT	*
		2715	*	* I\$WRK1 - 2 BYTES, CONTAINS CORE ADDR OF PRINT AREA LEFT BYTE	*
		2716	*	* I@WRK2 - 2 BYTES, CONTAINS VALUE FOR \$PRDEV 'CRT ONLY' COND	*
		2717	*	* ISSLLC - 1 BYTE, CONTAINS OUTPUT ELEMENT LENGTH CODE (LNG - 1)	*
		2718	*		*
		2719	*	OUTPUT -	*
		2720	*	* DISPLAYED ELEMENT AND/OR CURSOR CONTROL ON CRT DISPLAY UNIT	*
		2721		*****	
		2722	*		
		2723	*	ESTABLISH ADDRESSABILITY FOR PRINT ROUTINE (4TH VM PAGE)	
		2724	*		
		2725		*FZSP4B VPAGE 0	
3700		2726		ORG *,256,0 SET STARTING ADDRESS	
		2727	FZSP4B EQU *	START OF PROGRAM CODING	
3601		2728		ORG *-255 RESET IAR TO PAGE	
3700		2729		ORG *,256,0 * BOUNDARY ADDRESS	
		2730		USING *,@BR SET PAGE BASE ADDRESS	
3700		2731		ORG FZSP4B RESET STARTING ADDRESS	
		2732		*** END OF EXPANSION ***	
		2733	*		
		2734	*	PAGE ENTRY - ESTABLISH CRT IOCR EXECUTION CORE ADDRESS	
		2735	*		
3700 4C 01 D7 0D5B		2736	FZS800 MVC	FZS982+@OP1(,@BR),I\$WRK2(@CADDR) SET CRT EXECUTION CADDR	
		2737	*		
		2738	*	INITIALIZE FOR OUTPUT TO THE CRT DISPLAY UNIT	
		2739	*		
3705 7C 40 64		2740		MVI FZS4RM(,@BR),@DLNLG SET CRT RIGHT MARGIN PARAMETER	
		2741	*		
		2742	*	INITIALIZE THE ELEMENT PRINT PARAMETER LIST	
		2743	*		
3708 7C 40 E0		2744		MVI FZS4PF(,@BR),@PRINT SET FUNCTION FOR PRINT ONLY	
370B 4C 00 E1 0D56		2745		MVC FZS4PC(,@BR),I\$PARM-1(1) SET COUNT - ELEMENT CHAR COUNT	
3710 4C 01 E3 0D59		2746		MVC FZS4PA(,@BR),I\$WRK1(@CADDR) SET PRINT AREA CODE ADDRESS	
		2747	*		
		2748	*	TEST FOR AN ARITHMETIC ELEMENT - RETURN CURSOR IF ARITHMETIC	
		2749	*	ELEMENT LENGTH EXCEEDS OUTPUT LINE MARGIN	
		2750	*		
3715 5C 00 C6 E1		2751		MVC FZS4CC(,@BR),FZS4PC(1,@BR) SET PARAM = ELEMENT CHAR CNT	
3719 3D 12 0BA1		2752		CLI I\$SLLC,I@LCRV-1 IF CURR ELEMENT IS ARITHMETIC	
371D D0 01 BD		2753		BNE FZS960(,@BR) * LINK TO RTRN CURSOR ON COND	
		2754	*		
		2755	*	BRANCH TO APPROPRIATE ROUTINE DEPENDING ON CONTROL CODE	
		2756	*		
3720 4C 00 27 0D57		2757		MVC FZS805+@D1(,@BR),I\$PARM(1) MOVE CONTROL DISP TO JUMP INST	
3725 F2 87 00		2758	FZS805 J	*-* GO EXEC CONTROL CODE ROUTINE	
		2759	*		
		2760		*****	

## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 241
				2762			*****			
				2763	*		OUTPUT ROUTINE FOR PRINT CONTROL CODES 1, 9, 11, 13, 15			*
				2764			*****			
				2765	*					
				2766	*		DISPLAY THE FORMATTED ELEMENT ONLY (WHEN SIGNIFICANT)			
				2767	*					
3728	7D	00	E1	2768	FZS810	CLI	FZS4PC(,@BR),@ZERO			IF ELEMENT CHAR COUNT NOT ZERO
372B	F2	01	85	2769		JNE	FZS910			* GO DISPLAY ELEMENT ONLY,
372E	F2	87	88	2770		J	FZS950			* ELSE EXIT RTN W/O DISPLAYING
				2771	*					
3731	000000000000000000		373F	2772		DC	XL15'00'			PATCH SPACE 1-5
				2774			*****			
				2775	*		OUTPUT ROUTINE FOR PRINT CONTROL CODE 2			*
				2776			*****			
				2777	*					
				2778	*		ESTABLISH FULL PRINT ZONE OUTPUT FORMAT (ARITHMETIC ELEMENT)			
				2779	*					
3740	7C	12	C6	2780	FZS820	MVI	FZS4CC(,@BR),I@LFPZ			SET PARAM = FULL PRINT ZONE
3743	F2	87	18	2781		J	FZS836			BRANCH TO TEST LINE CAPACITY
				2782	*					
				2783			*****			



ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 242
		2785		*****	
		2786		* OUTPUT ROUTINE FOR PRINT CONTROL CODE 3	*
		2787		*****	
		2788		*	
		2789		* ESTABLISH PACKED PRINT ZONE OUTPUT FORMAT (ARITHMETIC ELEMENT) -	
		2790		* THIS ZONE WILL BE 6, 9, 12, 15, OR 18 CHARACTERS LONG DEPENDING ON	
		2791		* THE LENGTH OF THE ARITHMETIC ELEMENT TO BE PRINTED	
		2792		*	
3746	7C 04 C6	2793	FZS830 MVI	FZS4CC(,@BR),2*I@LPPZ-2	SET LENGTH ACCUN TO MINIMUM
		2794		*	* ELEMENT LENGTH LIMIT (4)
3749	5D 00 E1 C6	2795	FZS832 CLC	FZS4PC(,@BR),FZS4CC(1,@BR)	IF ELEMENT LENGTH WITHIN LIMIT
374D	F2 04 0A	2796		JNH FZS834	* BRANCH TO EXIT THIS LOOP
3750	5E 00 C6 DF	2797		ALC FZS4CC(,@BR),FZS4PZ(1,@BR)	ADD PACKED ZONE INCR TO ACCUM
3754	7D 10 C6	2798		CLI FZS4CC(,@BR),I@LFPZ-2	IF LENGTH ACCUM NOT MAXIMUM
3757	D0 82 49	2799		BL FZS832(,@BR)	* GO REPEAT ELEMENT LENGTH TEST
		2800		*	
375A	5E 00 C6 DE	2801	FZS834 ALC	FZS4CC(,@BR),FZS4B2(1,@BR)	ADJUST ACCUM TO MAKE PACKED
		2802		*	* PRINT ZONE FIELD LENGTH
		2803		*	
		2804		* TEST LINE CAPACITY TO CONTAIN CURRENT POINT ZONE FIELD - WHEN RIGHT	
		2805		* MARGIN IS EXCEEDED, LINE HAS CAPACITY FOR TED DATA ELEMENT BUT NOT	
		2806		* FOR THE ENTIRE PRINT ZONE ... IN THIS CASE, DISPLAY ELMEMENMT ONLY	
		2807		* AND RETURN THE CURSOR.	
		2808		*	
375E	4E 00 C6 03E2	2809	FZS836 ALC	FZS4CC(,@BR),\$CRPOS(1)	ADD PRINT ZONE LNG TO CURRENT
3763	7D 00 C6	2810	FZS838 CLI	FZS4CC(,@BR),*-*	* CURSOR POSITION - BRANCH
3766	F2 84 12	2811		JH FZS855	* IF RIGHT MARGIN IS EXCEEDED
		2812		*	
		2813		* LINE HAS CAPACITY FOR ENTIRE PRINT ZONE - DISPLAY ELEMENT AND SPACE	
		2814		* TO THE SPECIFIED ZONE POSITION	
		2815		*	
3769	4F 00 C6 03E2	2816	FZS840 SLC	FZS4CC(,@BR),\$CRPOS(1)	RESTORE CURRENT PRINT ZONE LNG
376E	5C 00 E1 C6	2817		MVC FZS4PC(,@BR),FZS4CC(1,@BR)	SET COUNT = CURR PRT ZONE LNG
3772	F2 87 3E	2818		J FZS910	GO DISPLAY ELEM & SPACE CURSOR
		2819		*	
		2820		*****	

## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 243
				2822			*****			
				2823	*		OUTPUT ROUTINE FOR PRINT CONTROL CODES 4, 12, 16			*
				2824			*****			
				2825	*					
				2826	*		TEST ELEMENT SIGNIFICANCE - RETURN CURSOR NO WHEN NOT SIGNIFICANT			
				2827	*					
3775	7D	00	E1	2828	FZS850	CLI	FZS4PC(,@BR),@ZERO			IF ELEMENT CHAR COUNT IS ZERO
3778	F2	81	20	2829		JE	FZS880			* GO RETURN THE CURSOR ONLY
				2830	*					
				2831	*		ELEMENT IS SIGNIFICANT - DISPLAY ELEMENT AND RETURN CURSOR			
				2832	*					
377B	7C	C0	E0	2833	FZS855	MVI	FZS4PF(,@BR),@PRETR			SET PRINT & CARR RETURN FUNC
377E	F2	87	32	2834		J	FZS910			GO DISPLAY ELEM AND RTRN CURSOR
				2836			*****			
				2837	*		OUTPUT ROUTINE FOR PRINT CONTROL CODE 5			*
				2838			*****			
				2839	*					
				2840	*		ESTABLISH FULL PRINT ZONE SPACING ONLY			
				2841	*					
3781	7C	12	E1	2842	FZS860	MVI	FZS4PC(,@BR),I@LFPZ			SET CO:AT R04 FLU *QM ZONE
3784	F2	87	03	2843		J	FZS875			BRANCH TO EXEC?TE SPACINS
				2845			*****			
				2846	*		OUTPUT ROUTINE FOR PRINT COHT4OL CODE 6			*
				2847			*****			
				2848	*					
				2849	*		ESTABLISH PACKED PRINT ZONE INCREMENT SPACING ONLY			
				2850	*					
3787	7C	03	E1	2851	FZS870	MVI	FZS4PC(,@BR),I@LPPZ			SET COUNT FOR PACKED ZONE INCR
				2852	*					
				2853	*		DISPLAY CURRENT ZONE, OR RETURN CURSOR IF END OF LINE IS HIT			
				2854	*					
378A	5C	00	C6	E1	2855	FZS875	MVC	FZS4CC(,@BR),FZS4PC(1,@BR)		SET PARAM FOR CURRENT ZONE LNG
378E	D0	87	BD	2856		B	FZS960(,@BR)			LINK TO RETURN CURSOR ON COND
3791	5D	00	C6	64	2857		CLC	FZS4CC(,@BR),FZS4RM(1,@BR)		IF CURSOS WAS NOT RETURNED
3795	F2	04	1B	2858		JNH	FZS910			* GO DISPLAY CURR ZONE SPACE
3798	F2	87	1E	2859		J	FZS950			* ELSE EXIT RTN W/O DISPLAYING
				2860	*					
				2861			*****			

## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	31/05/21	PAGE 244
		2863		*****			
		2864		* OUTPUT ROUTINE FOR PRINT CONTROL CODE 7			*
		2865		*****			
		2866		*			
		2867		* ESTABLISH CURSOR RETURN ONLY			
		2868		*			
379B	D2 02 E4	2869	FZS880	LA FZS4CR(, @BR), @XR			LOAD CURSOR RETURN PPL CADDR
379E	F2 87 15	2870		J FZS920			GO EXECUTE CURSOR RETURN
		2872		*****			
		2873		* OUTPUT ROUTINE FOR PRINT CONTROL CODE 8			*
		2874		*****			
		2875		*			
		2876		* RETURN CURSOR IF FULL PRINT ZONE EXCEEDS LINE CAPACITY			
		2877		*			
37A1	7C 12 C6	2878	FZS890	MVI FZS4CC(, @BR), I@LFPZ			SET PARAM FOR FULL PRINT ZONE
37A4	D0 87 BD	2879		B FZS960(, @BR)			LINK TO RETURN CLRSR ON COND
37A7	F2 87 0F	2880		J FZS950			GO EXIT DISPLAY ROUTINE
		2882		*****			
		2883		* OUTPUT ROUTINE FOR PRINT CONTROL CODE 10			*
		2884		*****			
		2885		*			
		2886		* RETURN CURSOR IF FULL PRINT ZONE EXCEEDS LINE CAPACITV			
		2887		*			
37AA	7C 12 C6	2888	FZS895	MVI FZS4CC(, @BR), I@LFPZ			SET PARAM FOR FULL PRINT ZONE
37AD	D0 87 BD	2889		B FZS960(, @BR)			LINK TO RETURN CURSOS ON COND
		2890		*			
		2891		*****			

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 245
		2893		*****	
		2894		* OUTPUT ROUTINE FOR PRINT CONTROL CODE 14	*
		2895		*****	
		2896		*	
		2897		* ESTABLISH FULL PRINT ZONE OUTPUT FORMAT (CHARACTER ELEMENT)	
		2898		*	
37B0	7C 12 E1	2899	FZS900	MVI FZS4PC(,@BR),I@LFPZ SET COUNT FOR FULL PRINT ZONE	
		2900		*	
		2901		* EXECUTE ELEMENT OUTPUT TO THE CRT DISPLAY UNIT	
		2902		*	
37B3	D2 02 E0	2903	FZS910	LA FZS4PL(,@BR),@XR LOAD DATA OUTPUT PPL CORE ADDR	
		2904		*	
37B6	D0 87 CE	2905	FZS920	B FZS980(,@BR) LINK TO EXECUTE CRT OUTPUT	
		2906		*	
		2907		* RETURN TO PRINT ROUTINE 3RD VM PAGE	
		2908		*	
37B9	C0 87 12D3	2909	FZS950	B I\$RTRN RETURN TO 3RD PRINT RTN PAGE	
		2910		*	
		2911		*****	

## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 246
		2913		*****	
		2914	*	DISPLAY UNIT CURSOR RETURN ROUTINE -	*
		2915	*	* RETURNS CURSOR WHEN SPECIFIED LENGTH PARAMETER (FZS4CC)	*
		2916	*	EXCEEDS THE CURRENT CRT DISPLAY LINE CAPACITY.	*
		2917		*****	
		2918	*		
37BD	74 08 DD	2919	FZS960 ST	FZS990+@OP1(,@BR),@ARR	STORE RETURN BRANCH ADDRESS
		2920	*		
		2921	*	TEST LINE CAPACITY TO CONTAIN CURRENT DISPLAY REGION LENGTH	
		2922	*		
37C0	4E 00 C6 03E2	2923	ALC	FZS4CC(,@BR),\$CRPOS(1)	ADD PRINT REGION LENGTH TO CURR
37C5	7D 00 64	2924	FZS970 CLI	FZS4RM(,@BR),*-*	* CURSOR POSITION - BRANCH IF
37C8	F2 02 0F	2925	JNL	FZS990	* RIGHT MARGIN NOT EXCEEDED
		2926	*		
		2927	*	RIGHT MARGIN EXCEEDED - RETURN DISPLAY UNIT CURSOR	
		2928	*		
37CB	D2 02 E4	2929	LA	FZS4CR(,@BR),@XR	LOAD CURSOR RETURN PPL CADDR
		2930	*		
		2931		*****	
		2932	*	DISPLAY UNIT OUTPUT INTERFACE -	*
		2933	*	* EXECUTES CRT DISPLAY OUTPUT AS SPECIFIED IN PRINT PARAMETER	*
		2934	*	* LIST REFERENCED BY REGISTER @XR.	*
		2935		*****	
		2936	*		
37CE	74 08 DD	2937	FZS980 ST	FZS990+@OP1(,@BR),@ARR	STORE RETURN BRANCH ADDRESS
		2938	*		
37D1	74 02 D9	2939	ST	FZS984(,@BR),@XR	STORE PPL CORE ADDRESS
37D4	C0 87 0000	2940	FZS982 B	*-*	LINK TO EXECUTE CRT IOCR
37D8		2941	FZS984 DS	CL(@CADDR)	CRT IOCS PARAMETER LIST CADDR
		2942	*		
		2943	*	RETURN TO CALLING ROUTINE	
		2944	*		
37DA	C0 87 0000	2945	FZS990 B	*-*	RETURN BRANCH
		2946	*		
		2947		*****	

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 247
			2949	*****	*****	
			2950	* PRINT EXECUTION ROUTINE CONSTANTS (4TH VM PAGE)	*	
			2951	*****	*****	
			2952	*		
37DE 02		37DE	2953	FZS4B2 DC	IL1'2' BINARY INTEGER +2	
			2954	*		
37DF 03		37DF	2955	FZS4PZ DC	AL1(I@LPPZ) LENGTH OF PACKED ZONE INCR	
			2956	*		
			2957	*****	*****	
			2958	* PRINT EXECUTION ROUTINE WORK AREAS (4TH VM PAGE)	*	
			2959	*****	*****	
			2960	*		
		3764	2961	FZS4RM EQU	FZS838+@Q CRT DISPLAY RIGHT MARGIN	
		37C6	2962	FZS4CC EQU	FZS970+@Q PRINT AREA CHARACTER COUNT	
			2963	*		
			2964	*FZS4PL PPL		
37E0 00		37E0	2965	FZS4PL EQU	* PPL ADDRESS	
37E1 00		37E0	2966	DC	AL1(*-*) FUNCTION REQUESTED	
37E2 0000		37E1	2967	DC	AL1(*-*) PRINT COUNT	
		37E3	2968	DC	AL2(*-*) DATA ADDRESS	
			2969	*** END OF EXPANSION ***		
		37E0	2971	FZS4PF EQU	FZS4PL+@PCTRL PRINT FUNCTION PARAMETER	
		37E1	2972	FZS4PC EQU	FZS4PL+@PRCNT PRINT AREA COUNT PARAMETER	
		37E3	2973	FZS4PA EQU	FZS4PL+@PDATA PRINT AKEA CADDR PARAMETER	
			2974	*		
			2975	*FZS4CR DPL	FUNC=@REYRN,CNT=@RTRNC	
37E4 80		37E4	2976	FZS4CR EQU	* PPL ADDRESS	
37E5 80		37E4	2977	DC	AL1(@RETRN) FUNCTION REQUESTED	
37E6 0000		37E5	2978	DC	AL1(@RTRNC) PRINT COUNT	
		37E7	2979	DC	AL2(*-*) DATA ADDRESS	
			2980	*** END OF EXPANSION ***		
			2981	*		
			2982	*****	*****	
			2983	*		
			2984	*** END OF PRINT EXECUTION ROUTINE CODING ***		
			2985	##### X'3800' #####		
			2986	* N O T Y E T S C A N N E D O R O B J C H E C K E D ! !		
			2987	##### X'4BFF' #####		
4BFF			2988	ORG	X'4BFF' T E M P ! ! !	

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	31/05/21	PAGE 248
		2990		*****			
		2991	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
		2992	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
		2993	*				*
		2994		*****			*
		2995	*	*STATUS			*
		2996	*	VERSION 1 MODIFICATION 0			*
		2997	*				*
		2998	*	*FUNCTION			*
		2999	*	* FZZVMP EXECUTION CAUSES ALL MODIFIED CORE VIRTUAL MEMORY PAGES			*
		3000	*	TO BE WRITTEN BACK TO DISK (PUSHED) OR ALL UNLOCKED CORE			*
		3001	*	VIRTUAL MEMORY PAGES TO BE LOADED INTO CORE (PULLED).			*
		3002	*	* OPERATION OF THIS ROUTINE DEPENDS UPON THE ENTRY POINT SELECTED			*
		3003	*	FOR EXECUTION -			*
		3004	*	* ENTRY POINT FZZVPS - ALL CORE VIRTUAL MEMORY PAGES REFER-			*
		3005	*	ENCED WITH A 'MODIFY' INDICATOR IN THE PAGING MODULE 'LOCK			*
		3006	*	AND READ ONLY' INDICATOR TABLE ARE WRITTEN INTO DISK			*
		3007	*	VIRTUAL MEMORY. THE 'MODIFY' INDICATOR IS UNSET IN THE			*
		3008	*	INDICATOR TABLE. THIS 'PUSH' IS AUTOMATICALLY ADJUSTED			*
		3009	*	TO PROCESS AN EXPANDED TABLE AND CORE PAGE REGION FOR			*
		3010	*	EXTENDED CORE CONFIGURATIONS.			*
		3011	*	* ENTRY POINT FZZVPL - ALL CORE VIRTUAL MEMORY PAGES REFER-			*
		3012	*	ENCED WITH A 'LOCK' INDICATOR IN THE PAGING MODULE 'LOCK			*
		3013	*	AND READ ONLY' INDICATOR TABLE ARE REPLACED WITH THE			*
		3014	*	CORRESPONDING PAGE FROM DISK VIRTUAL MEMORY. THIS 'PULL'			*
		3015	*	IS AUTOMATICALLY ADJUSTED TO PROCESS AN EXPANDED TABLE AND			*
		3016	*	CORE PAGE REGION FOR EXTENDED CORE CONFIGURATIONS.			*
		3017	*				*
		3018	*	*ENTRY POINTS			*
		3019	*	* ENTRY FZZVPS - FOR PERFORMING THE 'PUSH' OPERATION.			*
		3020	*	CALLING SEQUENCE IS			*
		3021	*	B IPGCAL			*
		3022	*	DC AL2(V\$VMPS)			*
		3023	*	WHERE THE ADDRESS CONSTANT PARAMETER DEFINES THE VIRTUAL			*
		3024	*	ADDRESS OF ENTRY POINT FZZVPS.			*
		3025	*	* ENTRY FZZVPL - FOR PERFORMING THE 'PULL' OPERATION.			*
		3026	*	CALLING SEQUENCE IS			*
		3027	*	B IPGCAL			*
		3028	*	DC AL2(V\$VMPL)			*
		3029	*	WHERE THE ADDRESS CONSTANT PARAMETER DEFINES THE VIRTUAL			*
		3030	*	ADDRESS OF ENTRY POINT FXXVPL.			*
		3031	*	* IN EACH CASE, EXECUTION IS SUBJECT TO THE INPUT CONDITIONS			*
		3032	*	DESCRIBED BELOW.			*
		3033	*				*
		3034	*	*INPUT			*
		3035	*	* \$EXFTR - 1 BYTE, FOR THE SYSTEM CORE EXTENSION FACTOR. THIS			*
		3036	*	CONTAINS THE NUMBER OF CORE PAGES (256-BYTE REGIONS) AVAILABLE			*
		3037	*	FOR GENERAL USE BEYOND THE 8K MINIMUM CONFIGURATION.			*
		3038	*	* PAGE INDICATOR TABLE - 10 BYTES (MINIMUM), FOR THE PAGING			*
		3039	*	MODULE 'LOCK AND READ ONLY' CORE VIRTUAL MEMORY INDICATORS.			*
		3040	*	THIS TABLE, WHICH IS EXPANDED TO (10+\$EXFTE-1) BYTES WHEN			*
		3041	*	\$EXFTR IS NON-ZERO, CONTAINS A SINGLE BYTE ENTRY CORRESPONDING			*
		3042	*	TO EACH CORE PAGE. BIT 6 (MASK X'02') IN EACH ENTRY INDICATES			*
		3043	*	THE MODIFICATION STATUS OF A CORE PAGE (1 = MODIFIED).			*
		3044	*	BIT 7 (MASK X'01') IN EACH ENTRY INDICATES THE LOCKED STATUS			*
		3045	*	OF A CORE PAGE (1 = LOCKED).			*



ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	31/05/21	PAGE 249
		3046	*	* PAGE REFERENCE TABLE - 256 BYTES, FOR THE PAGING MODULE CORE	*		
		3047	*	VIRTUAL MEMORY MAP. EACH BYTE IN THIS TABLE IS ASSOCIATED WITH	*		
		3048	*	A SPECIFIC VIRTUAL MEMORY PAGE, AND CONTAINS EITHER A VALUE OF	*		
		3049	*	ZERO OR THE NUMBER OF THE CORE PAGE CURRENTLY FILLED WITH THAT	*		
		3050	*	VIRTUAL MEMORY PAGE.	*		
		3051	*		*		
		3052	*	*OUTPUT	*		
		3053	*	* DISK VIRTUAL MEMORY - FOR ENTRY POINT FZZVPS ONLY, EACH CORE	*		
		3054	*	VIRTUAL MEMORY PAGE, FOR WHICH A 'PAGE MODIFY' BIT IS SET IS	*		
		3055	*	WRITTEN BACK TO DISK VIRTUAL MEMORY SO THAT DISK V.M. PAGES	*		
		3056	*	REFLECT THE CURRENT PROCESSING STATUS.	*		
		3057	*	* CORE VIRTUAL MEMORY - FOR ENTRY POINT FZZVPL ONLY, EACH CORE	*		
		3058	*	VIRTUAL MEMORY PAGE, FOR WHICH A 'PAGE LOCKED' BIT IS NOT SET,	*		
		3059	*	IS REPLACED WITH THE CORRESPONDING DISK VIRTUAL MEMORY PAGE	*		
		3060	*	SO THAT CORE V.M. PAGES REFLECT CURRENT DISK STATUS.	*		
		3061	*		*		
		3062	*	*EXTERNAL REFERENCES	*		
		3063	*	* \$DISKN - ENTRY POINT FOR THE SYSTEM PHYSICAL DISK IOCS.	*		
		3064	*	* \$WAITF - CORE ADDRESS OF 'WAIT' FUNCTION DISK PARAMETER LIST.	*		
		3065	*	* I\$RTRN - ENTRY POINT FOR PAGING MODULE V.M. RETURN CONTROL RTN.	*		
		3066	*	* \$EXFTR - 1 BYTE, FOR THE SYSTEM CORE EXTENSION FACTOR.	*		
		3067	*	* I\$CSXA - CORE ADDRESS OF 1ST BYTE IN CORE EXTENSION PAST 8K.	*		
		3068	*	* ISPLAT - CORE ADDRESS OF PAGE INDICATOR TABLE BASE ENTRY.	*		
		3069	*	* I\$PSTB - CORE ADDRESS OF PAGE REFERENCE TABLE BASE ENTRY.	*		
		3070	*		*		
		3071	*	*EXITS, NORMAL	*		
		3072	*	CONTROL IS ALWAYS PASSED TO THE PAGING ROUTINE AT ENTRY POINT.	*		
		3073	*	I\$RTRN (IPGRTN) FOR A RETURN TO THE CALLING PROGRAM.	*		
		3074	*		*		
		3075	*	*EXITS, ERROR	*		
		3076	*	N/A	*		
		3077	*		*		
		3078	*	*TABLES/WORK AREAS	*		
		3079	*	* DISK ADDRESS CONVERSION WORK AREAS - TWO 2-BYTE AREAS USED TO	*		
		3080	*	CONVERT LOGICAL DISK ADDRESSES TO PHYSICAL (A LA DL4ICS).	*		
		3081	*	* DISK PARAMETER LIST - 6 BYTES, FOR VIRTUAL PAGE READ/WRITE	*		
		3082	*	OPERATIONS.	*		
		3083	*		*		
		3084	*	*ATTRIBUTES	*		
		3085	*	* REUSABLE	*		
		3086	*	* NATURALLY RELOCATABLE	*		
		3087	*		*		
		3088	*	*CHARACTER CODE DEONENCY	*		
		3089	*	THE OPERATION OR THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*		
		3090	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*		
		3091	*		*		
		3092	*	*NOTES	*		
		3093	*	ERROR PROCEDURES	*		
		3094	*	NONE	*		
		3095	*		*		
		3096	*	REGISTER USAGE	*		
		3097	*	* REGISTER @BR IS TO CONTAIN THE CORE PAGE BASE ADDRESS	*		
		3098	*	ESTABLISHED THROUGH PAGING MODULE CONTROL FOR THE PAGE WHICH	*		
		3099	*	INCLUDES FZZVMP, AND IS RESTORED THROUGH THE PAGING MODULE.	*		
		3100	*	* REGISTER @XR IS NOT SAVED. IT IS USED IN FZZVMP FPR GENERAL	*		
		3101	*	PURPOSE INDEXING OPERATIONS.	*		

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 250	
					3102	*					*
					3103	*	SAVED/RESTORED AREAS				*
					3104	*	NONE				*
					3105	*					*
					3106	*	MODIFICATION CONSIDERATIONS				*
					3107	*	NONE				*
					3108	*					*
					3109	*	REQUIRED MODULES				*
					3110	*	* @SYSEQ - COMMON SYSTEM EQUATES				*
					3111	*	* @FXDEQ - SYSTEM NUCLEUS ADDRESSES AND INDICATOR EQUATES.				*
					3112	*	* \$B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.				*
					3113	*	* \$I\$EQU - INTERPRETER FIXED LOCATION ADDRESS EQUATES.				*
					3114	*	* \$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD PREC. ONLY)				*
					3115	*	* \$I@LEQ - INTERPRETER DARANETER EQUATES (FOR LNG PREC. ONLY)				*
					3116	*					*
					3117	*	OTHER				*
					3118	*	NONE				*
					3119	*	*****				*

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 251
					3121	*****				
					3122	* START OF VIRTUAL MEMORY PUSH/PULL EXECUTION ROUTINE				*
					3123	*****				
					3124	*				
					3125	* ESTABLISH VIRTUAL PAGE ADDRESSABILTY				
					3126	*				
					3127	*FZPGB VPAGE 0				
4C00					3128	ORG *,256,0	SET STARTING ADDRESS			
				4C00	3129	FZZPGB EQU *	START OF PROGRAM CODING			
4B01					3130	ORG *-255	RESET IAR TO PAGE			
4C00					3131	ORG *,256,0	* BOUNDARY ADDRESS			
				4C00	3132	USING *,@BR	SET PAGE BASE ADDRESS			
4C00					3133	ORG FZZPGB	RESET STARTING ADDRESS			
					3134	*** END OF EXPANSION ***				
					3135	*				
					3136	*****				

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 252
					3138	*				
					3139	*	ENTRY POINT FZZVPS - SET VIRTUAL PAGE PUSH FUNCTION.			
					3140	*				
				4C00	3141	FZZVPS EQU *	VM PUSH ROUTINE ENTRY POINT			
4C00	7C	02	BD		3142	MVI FZZDPL+@DCTRL(,@BR),@DPUT	SET DISK OUTPUT PARAMETER			
4C03	F2	87	03		3143	J FZZ005	GO PERFORM THE PUCH OPERATION			
					3144	*				
					3145	*	ENTRY POINT FZZVPL - SET VIRTUAL PAGE PULL FUNCTION.			
					3146	*				
				4C06	3147	FZZVPL EQU *	VM PULLH ROUTINE ENTRY POINT			
4C06	7C	01	BD		3148	MVI FZZDPL+@DCTRL(,@BR),@DGET	SET DISK OUTPUT PARAMETER			
					3150	*				
					3151	*	INITIALIZE PUSH/PULL ROUTINE FOR 8K SYSTEM ENVIRONMENT.			
					3152	*				
4C09	7C	0A	2B		3153	FZZ005 MVI FZZ020+@D1(,@BR),I@NCPG	SET MAX CORE PAGE COUNT FOR 8K			
4C0C	5C	01	BA B5		3154	MVC FZZHCA(,@BR),FZZSXA(@CADDR,@BR)	SET HIGH CORE ADDR FOR 8K			
					3155	*				
					3156	*	TEST FOR CORE AVAILABILITY BEYOND 8K - RE-INITIALIZE IF EXTENDED CORE			
					3157	*				
4C10	3D	00	043B		3158	CLI \$EXFTR,@ZERO	TEST FOR NULL CORE EXTENSION			
4C14	F2	81	0E		3159	JE FZZ010	BRANCH IF ONLY 8K SYSTEM CONFIG.			
					3160	*				
4C17	4E	00	2B 043B		3161	ALC FZZ020+@D1(,@BR),\$EXFTR(1)	ADD 1 LESS THAN EXTRA NO. OF			
4C1C	5F	00	2B B3		3162	SLC FZZ020+@D1(,@BR),FZZBN1(1,@BR)	* PAGES TO CORE PAGE COUNT			
4C20	4E	00	B9 043B		3163	ALC FZZHCA-1(,@BR),\$EXFTR(1)	SET EXTENDED SYSTEM HIGH CADDR			
					3164	*				
					3165	*****				

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 253
					3167	*				
					3168	*	ACCESS A CORE PAGE ENTRY IN THE PAGING MODULE 'LOCK AND READ ONLY'			
					3169	*	INDICATOR TABLE			
					3170	*				
4C25	C2	02	15E1		3171	FZZ010	LA I\$PLRT-1,@XR			LOAD CORE PAGE INDR TABLE BASE
4C29	E2	02	00		3172	FZZ020	LA *-*(,@XR),@XR			INCR POINTER TO CORE PAGE ENTRY
					3173	*				
					3174	*	TEST FOR PUSH OR PULL FUNCTION EXECUTION			
					3175	*				
4C2C	7D	01	BD		3176		CLI FZZDPL+@DCTRL(@BR),@DGET			IF DISK PARAM SET FOR INPUT
4C2F	F2	81	0C		3177		JE FZZ025			* BRANCH TO EXECUTE PAGE PULL
					3178	*				
					3179	*	PUSH FUNCTION - TEST THE CURRENTLY REFERENCED CORE PAGE INDICATOR			
					3180	*	FOR MODIFY BIT SET ON, AND PUSH THE CORE PAGE ONLY IF MODIFIED			
					3181	*				
4C32	B8	02	00		3182		TBN FZZLRT(@XR),FZZMDY			IF CORE PAGE IS NOT MODIFIED
4C35	F2	90	6A		3183		JF FZZ090			* GO DECREMENT CORE PAGE COUNT
4C38	BB	02	00		3184		SBF FZZLRT(@XR),FZZMDY			PAGE MODIFIED - SET INDICATOR
4C3B	F2	87	06		3185		J FZZ030			* OFF AND GO PERFORM PAGE PUSH
					3186	*				
					3187	*	PULL FUNCTION - TEST THE CURRENTLY REFERENCED CORE PAGE INDICATOR			
					3188	*	FOR LOCK BIT SET ON, AND PULL THE CORE PAGE ONLY IF NOT LOCKED			
					3189	*				
4C3E	B8	01	00		3190	FZZ025	TBN FZZLRT(@XR),FZZLOK			IF THE CORE PAGE IS LOCKED
4C41	F2	10	5E		3191		JT FZZ090			* GO DECREMENT CORE PAGE COUNT
					3192	*				
					3193	*	PUSH OR PULL CURRENTLY REFERENCED CORE PAGE - SEARCH THE PAGE			
					3194	*	REFERENCE TABLE TO DETERMINE THE ACTUAL VIRTUAL PAGE NUMBER			
					3195	*				
4C44	7C	FF	51		3196	FZZ030	MVI FZZ040+@D1(@BR),FZZBM1			SET VIRTUAL PAGE NO. = MINUS 1
4C47	C2	02	14CA		3197		LA I\$PGTB,@XR			LOAD PAGE REFERENCE TABLE BASE
4C4B	5E	00	51 B3		3198	FZZ035	ALC FZZ040+@D1(@BR),FZZBN1(1,@BR)			INCREMENT VIRTUAL PAGE NO.
4C4F	9D	00	00 2B		3199	FZZ040	CLC *-*(,@XR),FZZ020+@D1(1,@BR)			COMPARE REF TBL ENTRY W/ CORE
4C53	D0	01	4B		3200		BNE FZZ035(@BR)			* PAGE NO. AND LOOP IF NO MATCH
					3201	*				
					3202		*****			

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER	MOD	DATE	PAGE
					3204		*****				
					3205	*	CONVERT VIRTUAL PAGE NUMBER TO A PHYSICAL DISK ADDRESS				*
					3206		*****				
					3207	*					
					3208	*	ESTABLISH LOGICAL DISK ADDRESS IN THE DISK PARAMETER LIST				
					3209	*					
	4C56	7C	07	BE	3210	MVI	FZZDPL+@DCYL(,@BR),B@DVCY SET VIRTUAL MEMORY BASE CYL NO.				
	4C59	5C	00	BF 51	3211	MVC	FZZDPL+@DSAD(,@BR),FZZ040+@D1(1,@BR) SET RELATIVE SECTOR				
					3212	*	* ADDRESS EQUAL VIRT PAGE NO.				
					3213	*					
					3214	*	DETERMINE THE TRACK SECTOR COUNT (= LOGICAL SECTOR ADDRESS, MOD 24).				
					3215	*	INCREMENT THE CYLINDER/DISK/TRACK INDICATOR DURING EACH PASS THROUGH				
					3216	*	THE SUBTRACTION (DIVISION) LOOP.				
					3217	*					
	4C5D	5C	01	BC B8	3218	MVC	FZZCNT(,@BR),FZZCDT(@DADDR,@BR) INITLZ CYL/DISK/TRACK CNT				
	4C61	5F	01	BC B8	3219	FZZ050 SLC	FZZCNT(,@BR),FZZCDT(@DADDR,@BR) INCR CYL/DISK/TRACK COUNT				
	4C65	5F	00	BF B6	3220	SLC	FZZDPL+@DSAD(,@BR),FZZNST(1,@BR) DECR LOGICAL SECTOR ADDR				
	4C69	D0	02	61	3221	BNM	FZZ050(,@BR) REPEAT UNTIL SADDR IS NEGATIVE				
	4C6C	5E	00	BF B6	3222	ALC	FZZDPL+@DSAD(,@BR),FZZNST(1,@BR) RESTORE POSITIVE SADDR				
					3223	*					
					3224	*	THE DISK PARAMETER LIST NOW CONTAINS THE PHYSICAL SECTOR COUNT -				
					3225	*	THE CYLINDER CORRECTION COUNT CONTAINS THE INCREMENT WITH WHICH TO				
					3226	*	ADJUST THE LOGICAL CYLINDER ADDRESS, AND BITS 0 AND 1 OF THE DISK/				
					3227	*	TRACK INDICATOR BYTE ARE SET RESPECTIVELY TO THE CORRECT PHYSICAL				
					3228	*	DISK AND TRACK STATUS CONDITIONS.				
					3229	*					
					3230	*	CONVERT THE LOGICAL (BASE) CYLINDER ADDRESS TO A PHYSICAL ADDRESS				
					3231	*					
	4C70	5E	00	BE BB	3232	ALC	FZZDPL+@DCYL(,@BR),FZZCNT-1(1,@BR) ADJUST THE CYL ADDR				
					3233	*					
					3234	*	SHIFT SECTOR COUNT 2 BITS LEFT (MULTIPLY BY 4)				
					3235	*					
	4C74	5E	00	BF BF	3236	ALC	FZZDPL+@DSAD(,@BR),FZZDPL+@DSAD(1,@BR) SHIFT COUNT (2X)				
	4C78	5E	00	BF BF	3237	ALC	FZZDPL+@DSAD(,@BR),FZZDPL+@DSAD(1,@BR) SHIFT COUNT (4X)				
					3238	*					
					3239	*	SET THE SECTOR ADDRESS DISK (REMOVABLE OR FIXED) INDICATOR BIT				
					3240	*					
	4C7C	78	80	BC	3241	TBN	FZZCNT(,@BR),FZZIDM TEST INDICATOR DISK BIT				
	4C7F	F2	90	03	3242	JF	FZZ060 * AND BRANCH IF NOT EQUAL 1				
	4C82	7A	01	BF	3243	SBN	FZZDPL+@DSAD(,@BR),FZZSDM SET SADDR FOR FIXED DISK				
					3244	*					
					3245	*	SET THE SECTOR ADDRESS TRACK (UPPER OR LOWER) INDICATOR BIT				
					3246	*					
	4C85	78	40	BC	3247	FZZ060 TBN	FZZCNT(,@BR),FZZITM TEST INDICATOR TRACK BIT				
	4C88	F2	90	03	3248	JF	FZZ070 * AND BRANCH IF NOT EQUAL 1				
	4C8B	7A	80	BF	3249	SBN	FZZDPL+@DSAD(,@BR),FZZSTM SET SADDR FOR LOWER TRACK				
					3250	*					
					3251		*****				

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 255
					3253	*****				
					3254	* PERFORM READ/WRITE BETWEEN CORE PAGE AND DISK VIRTUAL MEMORY				*
					3255	*****				
					3256	*				
					3257	* CALCULATE THE AFFECTED CORE PAGE ACTUAL CORE ADDRESS				
					3258	*				
4C8E	5C	01	C2	BA	3259	FZZ070 MVC	FZZDPL+@DBFR2(,@BR),FZZHCA(@CADDR,@BR)			SET HIGH CORE ADDR
4C92	5F	00	C1	2B	3260	SLC	FZZDPL+@DBFR1(,@BR),FZZ020+@D1(1,@BR)			SUB CORE PAGE NO.
					3261	*				
					3262	* PERFORM THE CORE PAGE - VIRTUAL MEMORY DISK OPERATION				
					3263	*				
4C96	D2	02	BD		3264	LA	FZZDPL(,@BR),@XR			LOAD PARAMETER LIST CORE ADDR
4C99	74	02	A1		3265	ST	FZZ080(,@BR),@XR			STORE DPL CORE ADOR FOR CALL
4C9C	C0	87	0025		3266	B	\$DISKN			LINK TO READ/WRITE THE CORE PAGE
4CA0				4CA1	3267	FZZ080 DS	CL(@CADDR)			PARAMETER LIST CORE ADDRESS
					3269	*				
					3270	* SET NEXT CORE PAGE PROCESSING - EXIT IF NO MORE CORE PAGES				
					3271	*				
4CA2	5F	00	2B	B3	3272	FZZ090 SLC	FZZ020+@D1(,@BR),FZZBN1(1,@BR)			DECR THE CORE PAGE NUMBER
4CA6	D0	84	25		3273	BP	FZZ010(,@BR)			GO PROCESS NEW PAGE UNLESS ZERO
					3274	*				
					3275	* EXIT - RETURN TO THE CALLING ROUTINE				
					3276	*				
4CA9	C0	87	0025		3277	B	\$DISKN			LINK TO WAIT I/O COMPLETED
4CAD	057F			4CAE	3278	DC	AL(@CADDR)(\$WAITF)			'WAIT' FUNCTION PARAM CADDR
					3279	*				
4CAF	C0	87	12D3		3280	B	I\$RTRN			RETURN TO CALLING ROUTINE
					3281	*				
					3282	*****				



ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 256
					3284	*****		
					3285	* VIRTUAL MEMORY PUSH/PULL ROUTINE CONSTANTS		*
					3286	*****		
					3287	*		
4CB3	01			4CB3	3288	FZZBN1 DC	IL1 '1'	BINARY INTEGER +1
					3289	*		
4CB4	2000			4CB5	3290	FZZSXA DC	AL (@CADDR) (I\$CSXA)	CORE EXTENSION STARTING ADDRESS
					3291	*		
4CB6	18			4CB6	3292	FZZNST DC	AL1 (@DTRSZ)	NO. OF SECTORS PER DISK TRACK
4CB7	FFC0			4CB8	3293	FZZCDT DC	XL (@DADDR) 'FFC0'	CYLINDER/DISK/TRACK DECREMENT
					3295	*****		
					3296	* VIRTUAL MEMORY PUSH/PULL ROUTINE WORK AREAS		*
					3297	*****		
					3298	*		
4CB9				4CBA	3299	FZZHCA DS	CL (@CADDR)	HIGHEST AVAILABLE CADDR + 1
					3300	*		
4CBB				4CBC	3301	FZZCNT DS	CL (@DADDR)	CYLINDER/DISK/TRACK COUNTER
					3302	*		
					3303	*FZZDPL DPL	CNT-1	VM I/O DISK PARAMETER LIST
				4CBD	3304	FZZDPL EQU	*	DISK PARAMETER LIST
4CBD	00			4CBD	3305		DC AL1 (*-*)	REQUESTED FUNCTION
4CBE	00			4CBE	3306		DC AL1 (*-*)	CYLINDER ADDRESS
4CBF	00			4CBF	3307		DC AL1 (*-*)	HEAD/SECTOR/DRIVE/DISK SPEC
4CC0	01			4CC0	3308		DC AL1 (1)	SECTOR COUNT
4CC1	0000			4CC2	3309		DC AL2 (*-*)	BUFFER ADDRESS
					3310	***	END OF EXPANSION ***	
					3312	*****		
					3313	* VIRTUAL MEMORY PUSH/PULL ROUTINE EQUATES REFERENCING CONSTANTS		*
					3314	*****		
					3315	*		
				00FF	3316	FZZBM1 EQU	X 'FF'	BINARY INTEGER -1
					3317	*		
				0000	3318	FZZLRT EQU	0	DISP FOR PAGE INDR TABLE ENTRY
				0001	3319	FZZLOK EQU	X '01'	CORE PAGE INDICATOR LOCK MASK
				0002	3320	FZZMDY EQU	X '02'	CORE PAGE INDICATOR MODIFY MASK
					3321	*		
				0080	3322	FZZIDM EQU	X '80'	INDICATOR DISK BIT MASK
				0040	3323	FZZITM EQU	X '40'	INDICATOR TRACE BIT MASK
				0001	3324	FZZSDM EQU	X '01'	SECTOR ADDR DISK BIT MASK
				0080	3325	FZZSTM EQU	X '80'	SECTOR ADDR TRACK BIT MASK
					3326	*		
					3327	*	END OF VIRTUAL MEMORY PUSH/PULL ROUTINE CODING	*****
					3328	*		

## DLFPRT - LINE PRINTER ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	31/05/21	PAGE 257
		3330		*****			
		3331	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
		3332	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
		3333	*				*
		3334		*****			*
		3335	*	STATUS			*
		3336	*	VERSION 1 MODIFICATION 0			*
		3337	*				*
		3338	*	FUNCTION			*
		3339	*	* DLFPRT EXECUTION CAUSES DATA OUTPUT AND/OR CARRIER POSITIONING			*
		3340	*	ON THE SYSTEM PRINT DEVICE UNDER CONTROL OF CODES RECEIVED FROM			*
		3341	*	THE CALLING ROUTINE, PRINTING IS DONE BIDIRECTIONALLY			*
		3342	*	* THE FOLLOWING ACTIONS ARE PERFORMED DEPENDING ON THE CODE AND			*
		3343	*	CARRIER POSITION:			*
		3344	*	* INDEX, PRINT AND INDEX & TAB, PRINT AND INDEX			*
		3345	*	* INPUT CODES			*
		3346	*	* PRINT X'40' WILL CAUSE THE DATA TO BE PRINTED TO			*
		3347	*	BE MOVED INTO THE LINE PRINTER BUFFER			*
		3348	*	* PRINT & RETRN X'C0' WILL CAUSE THE DATA TO BE MOVED INTO			*
		3349	*	THE BUFFER, AND THE CONTENTS PRINTED			*
		3350	*	* CARRAGE RETRN X'80' WILL CAUSE AN INDEX IF THE BUFFER IS			*
		3351	*	EMPTY OR THE BUFFER PRINTED IF NOT			*
		3352	*				*
		3353	*	ENTRY POINTS			*
		3354	*	THIS ROUTINE HAS A SINGLE CALLING ENTRY POINT - DLFPRT - WHOSE			*
		3355	*	FUNCTION IS DEFINED ABOVE. THE CALLING SEQUENCE IS:			*
		3356	*	B I\$LDXR			*
		3357	*	DC AL2(V\$LPRT)			*
		3358	*	WHERE THE ADDRESS CONSTANT PARAMETER DEFINES THE VIRTUAL ADDRESS			*
		3359	*	OF ENTRY POINT DLFPRT.			*
		3360	*				*
		3361	*	INPUT			*
		3362	*	* \$PRPOS - 1 BYTE CARRIER POSITION RELATIVE TO HARDWARE LEFTMGN			*
		3363	*	* \$LMRGN - 1 BYTE SOFTWARE LEFT MARGIN INDICATOR			*
		3364	*				*
		3365	*	OUTPUT			*
		3366	*	* PRINTED OUTPUT AND CARRIER POSITIONING			*
		3367	*	* \$PRPOS - 1 BYTE 'DUMMY' CARRIER POSITION INDICATING WHERE THE			*
		3368	*	CARRIER SHOULD BE. SET EQUAL TO \$LMRGN AFTER PRINTING.			*
		3369	*	* \$BUFPT - 1 BYTE POINTS AT NEXT AVAIL BYTE IN LINE PRINT BUFFER			*
		3370	*	* \$LPRP3 - 1 BYTE LINE PRINTER INDICATORS			*
		3371	*	* 3LPRI0 - 2 BYTES ONE FOR BUFFER INCREMENT ONE FOR PDAR DISP.			*
		3372	*				*
		3373	*	EXTERNAL REFERENCES			*
		3374	*	* V\$LPRT2 - VIRTUAL ENTRY SECOND PAGE OF LINE PRINTER ROUTINE			*
		3375	*	* V\$LPRTB - VIRTUAL ADDRESS OF THE LINE PRINTER BUFFER			*
		3376	*	* I\$LDXR - ENTRY POINT FOR PAGING MODULE V.M. LOAD XR ROUTINE			*
		3377	*	* \$LPRI0 - ENTRY POINT FOR PAGING MODULE V.M. CONVERT ADDRESS			*
		3378	*				*
		3379	*	EXITS, NORMAL			*
		3380	*	EXIT IS TO THE CALLING ROUTINE VIA A BRANCH TO THE V.M. PAGING			*
		3381	*	ROUTINE.			*
		3382	*				*
		3383	*	EXITS, ERROR			*
		3384	*	NONE			*
		3385	*				*

## DLFPRT - LINE PRINTER ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	31/05/21	PAGE 258
		3386	*	TABLES/WORKAREAS			*
		3387	*	N/A			*
		3388	*				*
		3389	*	ATTRIBLTES			*
		3390	*	NATURALLY RELOCATABLE AND REUSABLE			*
		3391	*				*
		3392	*	CHARACTLR CODE DEPENDENCY			*
		3393	*	THE OPERATION OF THIS MODULE DEPENDS UPON AN INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET WHICH IS EQUIVALENT TO THE			*
		3394	*	ONE USED AT ASSEMBLY TIME.			*
		3395	*				*
		3396	*				*
		3397	*	NOTES			*
		3398	*	ERROR PROCEDLRES			*
		3399	*	IF A PRINTER UNIT CHECK OCCURES. THE LINE IN WHICH THE CHECK			*
		3400	*	OCCURED WILL BE REPRINTED			*
		3401	*				*
		3402	*	REGISTER USAGE			*
		3403	*	REGISTER 1 (@BR) IS USED AS A BASE REGISTER FOR DFPRNT			*
		3404	*	REGISTER 2 (@XR) IS USED AS A BASE REGISTER FOR: THE FIRST			*
		3405	*	PAGE OF DLFPRT, LINE PRINTER BUFFER, OR IN THE CASE OF A UNIT			*
		3406	*	CHECK, THE PRINTER ERROR HANDELING ROUTINE 'DFPNDX'.			*
		3407	*				*
		3408	*	SAVED/RESTORED AREAS			*
		3409	*	NONE			*
		3410	*				*
		3411	*	MODIFICATION CONSIDERATIONS			*
		3412	*	CHANGES TO EITHER DLFPRT OR DFPRNT MAY DIRECTLY AFFECT THE			*
		3413	*	INTERFACE BETWEEN THE TWO MODULES.			*
		3414	*				*
		3415	*	REQUIRED MODULES			*
		3416	*	@SYSEQ			*
		3417	*	@FXDEQ			*
		3418	*	@HDWEQ			*
		3419	*	\$V\$EQU			*
		3420	*	\$I\$EQU			*
		3421	*	DFPRNT			*
		3422	*				*
		3423	*	OTHER			*
		3424	*	NONE			*
		3425	*	*****			*

## DLFPRT - LINE PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 259
					3427	*****				
	4D00				3428	ORG	*,256,0			SET STARTING ADDRESS
				2800	3429	USING	DFPASE,@BR			SET PAGE BASE ADDRESS - DFPRNT
				4D00	3430	USING	DLFPRT,@XR			SET PAGE BASE ADDRESS
					3431	*				
				4D00	3432	DLFPRT EQU	*			ENTRY BIDIR PRINT
	4D00	7C	87	BC	3433	MVI	DFP330+@Q(,@BR),@UCB			SET BRANCH TO LINE PRINTER PAGE
	4D03	B4	02	66	3434	ST	DLF155+@OP1(,@XR),@XR			SAVE XR
	4D06	3A	40	03E4	3435	SBN	\$LPRP3,@PRINT			SET LINE PRINTER FLAG
	4D0A	2C	01	144A D7	3436	MVC	I\$VADR,DLFVD1(@VADDR,@XR)			GET PRINTER BUFFER VADDR
	4D0F	C0	87	1349	3437	B	I\$MDFY			LOAD BUFFER & SET PAGE MDFY BIT
	4D13	8C	01	D9 144C	3438	MVC	BUFADR(2,@XR),I\$CADR			SAVE BUFFER ADDR
				4D18	3439	DLF050 EQU	*			PROCESS PRINTER UNIT CHECK
	4D18	7C	25	BD	3440	MVI	DFP330+@D1(,@BR),DENTRY			SET ENTRY DISPLACEMENT
	4D1B	BC	87	A9	3441	MVI	DLF360+@Q(,@XR),@UCB			FORCE RETURN ENTRY
	4D1E	6C	02	BA F6	3442	MVC	DFP333(3,@BR),DLFEOR(,@XR)			SET DLFPRT ERROR ENTRY
					3443	*				
	4D22	D0	87	A2	3444	B	DFP280(,@BR)			GO CHECK FOR PREV. ERROR
					3446	*****				
					3447	*				
					3448	*	FIND FUNCTION			
					3449	*				
					3450	*****				
				4D25	3451	DLF100 EQU	*			RETURN FROM ERROR CHECK
	4D25	BC	80	A9	3452	MVI	DLF360+@Q(,@XR),@NOP			RESET ENTRY INDICATOR
	4D28	78	40	F5	3453	TBN	DLFIST+@PCTRL(,@BR),@PRINT			IS OP A PRINT ?
	4D2B	F2	90	4A	3454	JF	DLF170			CHECK IF BUFFER FULL
					3455	*****				
					3456	*				
					3457	*	ENTRY TO FILL BUFFER			
					3458	*				
					3459	*****				
	4D2E	39	01	03E4	3460	TBF	\$LPRP3,@INDEX			TEST DUMMY PRINT
	4D32	F2	90	0A	3461	JF	DLF140			SKIP IF IN USE
	4D35	3A	01	03E4	3462	SBN	\$LPRP3,@INDEX			SET DUMMY PRINT POS. USED
	4D39	0C	00	03E5 03C2	3463	MVC	\$LPROS(1),\$PRPOS			SAVE TRUE POSITION
				4D3F	3464	DLF140 EQU	*			UPDATE BUFFER POINTER
					3465	*				
					3466	*****				
					3467	*				
	4D3F	1E	00	03E3 F6	3468	ALC	\$BUFPT,DLFIST+@PRCNT(1,@BR)			ADD NEXT COUNT TO BUFFER PTR
	4D44	1E	00	03C2 F6	3469	ALC	\$PRPOS(1),DLFIST+@PRCNT(,@BR)			UPDATE HEAD POSITION
					3470	*				
					3471	*	INCREMENT BUFFER POINTER			
					3472	*				
	4D49	2C	01	144A ED	3473	MVC	I\$VADR,DLFPCH(@VADDR,@XR)			V.M. PATCH PAGE ENTRY ADDR 1-5
	4D4E	C0	87	1358	3474	DLF143 B	I\$CVAD			LOAD PATCH PAGE 1-5
	4D52	8C	01	5A 144C	3475	MVC	DLF145+@OP1(@CADDR,@XR),I\$CADR			MOVE CADDR TO BRANCH 1-5
	4D57	C0	87	0000	3476	DLF145 B	*-*			1-5
					3477	*				
					3478	*	MOVE DATA TO BUFFER			
					3479	*				
	4D5B	B5	02	D9	3480	DLF146 L	BUFADR(,@XR),@XR			XR - BUFFER CADDR
	4D5E	8C	00	00 0000	3481	DLF150 MVC	*-*(@VQ,@XR),*-*			MOVE DATA INTO BUFFER
					3482	*				

## DLFPRT - LINE PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 260
	4D63	C2	02	0000	3483	DLF155	LA *-*,@XR	RESTORE DLFPRT BASE ADDR
					3484	*		
					3485	*	TEST FOR CARRAGE RETURN	
					3486	*		
	4D67	7D	C0	F5	3487		CLI DLFIST+@PCTRL(,@BR),@PRETR	TEST CARRAGE RETURN ON
	4D6A	F2	01	4C	3488		JNE DLF175	JUMP TO RETURN IF NO C.R.
				4D6D	3489	DLF160	EQU *	LOAD PAGE2 LINE PRINTER
	4D6D	7C	88	BD	3490		MVI DFP330+@D1(,@BR),DERROR	SET ERROR ENTRY DISP.
	4D70	2C	01	144A EB	3491	DLF165	MVC I\$VADR,DLFVD2(@VADDR,@XR)	VADDR VLPRT2
	4D75	E0	87	93	3492		B DLF400(,@XR)	LOAD BASE
				4D78	3494	DLF170	EQU *	CHECK IF BUFFER EMPTY
	4D78	3D	00	03E3	3495		CLI \$BUFPT,@ZERO	IS BUFFER EMPTY ?
	4D7C	E0	01	6D	3496		BNE DLF160(,@XR)	GO TO PRINT EXIT
	4D7F	7C	01	DE	3497		MVI DLFPCF(,@BR),@INDEX	SET INDEX ONLY
	4D82	7C	87	A0	3498		MVI DFP270+@Q(,@BR),@UCB	FORCE RETURN
	4D85	D0	87	92	3499		B DFP240(,@BR)	GO DO I/O
					3501	*		
					3502	*	NO ERROR, CHECK FOR PREVIOUS ERROR	
					3503	*		
	4D88	F2	00	1D	3504	DLF350	JC DLF360,*-*	JUMP NO PREVIOUS ERROR
	4D89				3505		ORG DLF350+@Q	* INITIALIZE
	4D89	87			3506		DC AL1(@UCB)	* TO INDICATE
	4D8B				3507		ORG DLF350+@INST3	* NO PREVIOUS PRINTER ERROR
	4D8B	BC	87	89	3508		MVI DLF350+@Q(,@XR),@UCB	RESET ERROR INDICATOR
	4D8E	2C	01	144A E3	3509	DLF355	MVC I\$VADR,DLFRTY(@VADDR,@XR)	VADDR RETRY ENTRY VLPRT2
				4D93	3510	DLF400	EQU *	PREPARE TO EXIT LINE PTR PAGE1
	4D93	3C	80	12B6	3511		MVI I\$LBFR,@NOP	FORCE LINE PRINTER UNLOCK
	4D97	C0	87	1358	3512		B I\$CVAD	LOAD LINE PRINTER PAGE2
	4D9B	8C	01	A7 144C	3513		MVC DLF425+@OP1(@CADDR,@XR),I\$CADR	MOVE CADDR TO BR
	4DA0	C0	87	1354	3514		B I\$LOCK	LOCK PAGE VLPRT2 1-5
	4DA4	C0	87	0000	3515	DLF425	B *-*	BRANCH TO PAGE2
	4DA8	E0	00	25	3517	DLF360	BC DLF100(,@XR),*-*	FORMAT NEXT LINE / GO TO ENTRY
	4DA9				3518		ORG DLF360+@Q	* INITIALIZE
	4DA9	80			3519		DC AL1(@NOP)	* TO FORMAT
	4DAB				3520		ORG DLF360+@INST3	* NEXT LINE TO BE PRINTED
	4DAB	2C	01	144A EF	3521		MVC I\$VADR,DLFPC1(@VADDR,@XR)	V.M. PATCH PAGE ENTRY ADDR 1-5
	4DB0	E0	87	4E	3522	DLF375	B DLF143(,@XR)	BRANCH TO MV CADDR TO BRANCH 1-5
					3524	*****		
					3525	***** RETURN TO CALLER *****		
					3526	*****		
				4DB3	3527	RETURN	EQU *	LINE PRINTER RETURN AREA
	4DB3	0C	00	03C2 03C1	3528		MVC \$PRPOS(1),\$LMRGN	SET DUMMY POSITION LEFT MGN
				4DB9	3529	DLF175	EQU *	RETURN FROM DLFPRT
	4DB9	7C	80	BC	3530		MVI DFP330+@Q(,@BR),@NOP	RESET BRANCH TO LINR PRINTER
	4DBC	7C	80	A0	3531		MVI DFP270+@Q(,@BR),@NOP	RESET DFPRNT EXIT
	4DBF	6C	02	BA F3	3532		MVC DFP333(3,@BR),DFPEOR(,@XR)	RESTORE DFPRNT ERROR TEST
	4DC3	7C	11	E0	3533		MVI DLFPCF+2(,@BR),@TBLIX	RESTORE MATRIX PRINTER END
	4DC6	3B	40	03E4	3534		SBF \$LPRP3,@PRINT	RESET LINE PRINTER FLAG
	4DCA	D0	87	CA	3535		B DFP300(,@BR)	RETURN TO CALLER
					3536	*		
					3537	*****		
	4DCD				3538	DLFRPE	EQU *	PRINTER UNIT CHECK ENTRY

## DLFPRT - LINE PRINTER ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 261

4DCD	C0 87 1330		3539	B	I\$LDXR	BR TO FORCE DLFPRT TO BE MOST	
			3540	*		* RECENTLY USED PAGE	
4DD1	4D00	4DD2	3541	DC	AL2(V\$LPRT)	DLFPRT VADDR	
4DD3	D0 87 D3		3542	B	DFPRPE-DFPRNT(,@BR)	GO PROCESS LOAD ERP SECTION	
			3543	*			
			3544	*	*****		
4DD6	4F00	4DD7	3545	DLFVD1 DC	AL(@VADDR)(V\$LPRB)	LINE PRINTER BUFFER PAGE	
4DD8	0000	4DD9	3546	BUFADR DC	XL2'00'	SAVED BUFFER ADDR	
			3547	*			
4DDA	0000	4ddb	3548	DFPWIDTH DC	XL2'00'	LINE WIDTH	
4DDC	00	4DDC	3549	DFPRES DC	XL1'00'	LINE COUNT	
4DDD	0000	4DDE	3550	BUFRWK DC	XL2'00'	BUFFER POINTER	
4DDF	00	4DDF	3551	DLFBPT DC	XL1'00'	BUFFER INCREMENT	
			3552	*			
4DE0	0025	4DE1	3553	DLFMAR DC	AL2(DLF500-VLPRT2)	DISPLACENENT TO FORMAT LINE	
4DE2	4E49	4DE3	3554	DLFRTY DC	AL2(V\$LPRT2+DLF700-VLPRT2)	RETRY ENTRY POINT	
			3555	*			
4DE4	00	4DE4	3556	DFPPOS DC	XL1'00'	CHARACTER POSITION ON LINE	
4DE5	8080C00001	4DE9	3557	LPRCMD DC	XL5'8080C00001'	LINE PRINTER CMDS.	
4DEA	4E00	4DEB	3558	DLFVD2 DC	AL2(V\$LPRT2)	LINE PRINTER PAGE2	
		004E	3559	DLFX4E EQU	X'4E'	VLPRT2 LOCK BIT	1-5
		0053	3560	DLFX53 EQU	X'53'	VLPRT3 LOCK BIT	1-5
		0090	3561	DLTABL EQU	X'90'	TAB LEFT AND CHAIN	
4DEC	5391	4DED	3562	DLFPCH DC	AL2(V\$PCH2+DLF400-@D1-DLFPRT)	PATCH PAGE ENTRY ADDR	1-5
4DEE	53B6	4DEF	3563	DLFPC1 DC	AL2(V\$PCH2+DLF175-@DD2-DLFPRT)	PATCH PAGE ENTRY ADDR	1-5
4DF0	00	4DF0	3564	DLFSWC DC	XL1'00'	RETURN CARRIAGE SWITCH	1-5
		00A0	3565	DLTABR EQU	X'A0'	TAB RIGHT AND CHAIN	
		0088	3566	DERROR EQU	DLF350-DLFPRT	ERROR CHECK ENTRY DISP.	
		0025	3567	DENTRY EQU	DLF100-DLFPRT	ENTRY RETURN DISP.	
		0001	3568	DLFRTN EQU	X'01'	RETURN CARRIAGE INDICATOR	1-5
			3569	*			
			3570	*	INSTRUCTION MODIFICATION TP DFPRNT AT DFP335		
			3571	*			
4DF1	D1 E0 D3		3572	TIO	DFPRPE-DFPRNT(,@BR),@PERR	FORCE BRANCH TO DFPRNT ERROR	
		4DF3	3573	DFPEOR EQU	*-1	LAST BYTE OF FORCE DFPRNT ERROR	
4DF4	E1 E0 CD		3574	TIO	DLFRPE(,@XR),@PERR	FORCE BRANCH TO DLFPRT ERROR	
		4DF6	3575	DLFEOR EQU	*-1	LAST BYTE DLFPRT FORCE ERROR	
			3576	*	*****		
			3577	*	END V\$LPRT	*****	
			3578	*	*****		



## DLFPRT - LINE PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 262
					3580		*****			
					3581	*				
					3582	*	ENTRY TO FORMAT PRINT LINE			
					3583	*				
					3584		*****			
4E00					3585	ORG	*,256,0 SET STARTING ADDRESS			
				2800	3586	USING	DFPASE,@BR SET PAGE BASE ADDRESS - DFPRNT			
				4D00	3587	USING	DLFPRT,@XR SET PAGE BASE ADDRESS			
				4E00	3588	VLPRT2 EQU	*			
4E00	2C	01	144A	D7	3589	MVC	I\$VADR,DLFVD1(@VADDR,@XR) GET BUFFER ADDR			
4E05	C0	87	1354		3590	B	I\$LOCK LOCK PRINT BUFFER			
4E09	8C	01	D9	144C	3591	MVC	BUFADR(2,@XR),I\$CADR SAVE LINE PRINTER BUFFER CADDR			
4E0E	8C	01	DE	144C	3592	MVC	BUFRWK(2,@XR),I\$CADR SAVE BUFFER ADDRESS			
					3593		*****			
					3594	*				
					3595	*	DETERMINE ANY MARGIN COMPUTATION REQUIRED			
					3596	*				
					3597		*****			
4E13	8C	00	DC	03E3	3598	MVC	DFPRES(1,@XR),\$BUFPT SAVE COUNT			
4E18	8C	00	DB	03C0	3599	MVC	DFPWT(1,@XR),\$RMRGN SET RIGHT MARGIN VALUE			
4E1D	8F	00	DB	03C1	3600	SLC	DFPWT(1,@XR),\$LMRGN CALCULATE WIDTH			
4E22	F2	87	04		3601	J	DLF525 CONTINUE			
					3602	*				
				4E25	3603	DLF500 EQU	* FORMAT LINE			
4E25	AE	01	DE	DB	3604	ALC	BUFRWK(2,@XR),DFPWT(,@XR) GET NEXT PDAR ADDR			
				4E29	3605	DLF525 EQU	*			
4E29	AD	00	DB	DC	3606	CLC	DFPWT(1,@XR),DFPRES(,@XR) COMPARE WIDTH TO LINE LNTH			
4E2D	F2	02	0C		3607	JNL	DLF550 JUMP LENGTH < WIDTH			
					3608		*****			
					3609	*				
					3610	*	COMPUTE MARGIN AND FORMAT DATA			
					3611	*				
					3612		*****			
4E30	AF	00	DC	DB	3613	SLC	DFPRES(1,@XR),DFPWT(,@XR) NEXT LINE = RESIDUAL			
4E34	2C	00	03E3	DB	3614	MVC	\$BUFPT(1),DFPWT(,@XR) SET NEW LINE - WIDTH			
4E39	F2	87	08		3615	J	DLF600 GO TO FORMAT NEXT LINE			
					3616	*				
					3617	*	COUNT < WIDTH			
					3618	*				
				4E3C	3619	DLF550 EQU	*			
4E3C	2C	00	03E3	DC	3620	MVC	\$BUFPT(1),DFPRES(,@XR) \$BUFPT RESIDUAL			
4E41	7C	87	A0		3621	MVI	DFP270+@Q(,@BR),@UCB FORCE LINE PRINT EXIT			
					3622	*				
				4E44	3623	DLF600 EQU	* FORMAT LINE			
4E44	8C	00	DF	03E3	3624	MVC	DLFBPT(1,@XR),\$BUFPT SAVE BUFFER POINTER			
				4E49	3625	DLF700 EQU	* PRINT RETRY ENTRY POINT			
4E49	B1	E4	DE		3626	LIO	BUFRWK(,@XR),@PDAR SET DATA ADDR			
4E4C	6C	04	E2	E9	3627	MVC	DFPPCO(5,@BR),LPRCMD(,@XR) SET LINE PRINTER CMDS.			
					3628	*				
					3629	*	COMMON MARGIN ENTRY			
					3630	*				
4E50	7C	00	9E		3631	MVI	DFP260-DFPRNT+@D1(,@BR),@ZERO SET TO PRINT RIGHT			
4E53	8C	00	E4	03E5	3632	MVC	DFPPOS(1,@XR),\$LPROS GET ACTUAL POSITION			
4E58	0C	00	03E5	03C1	3633	MVC	\$LPROS(1),\$LMRGN SET REFERENCE			
4E5E	0E	00	03E5	03E3	3634	ALC	\$LPROS(1),\$BUFPT UPDATE PRINT POSITION			
					3635	*				



## DLFPRT - LINE PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 263	
4E64	1F	00	03E3 E7	3636	SLC	\$BUFPT(1),DLF001(,@BR)	COUNT LESS ONE		
4E69	4C	00	E1 03E3	3637	MVC	DLFPCF+3(1,@BR),\$BUFPT	MOVE DATA COUNT TO PCF		
4E6E	2D	00	03C1 E4	3638	CLC	\$LMRGN(1),DFPPOS(,@XR)	AT LEFT MARGIN ?		
4E73	F2	81	61	3639	JE	DLF950	JUM IF AT LEFT MARGIN		
				3641	*****				
				3642	*				
				3643	CALCULATE TAB				
				3644	***	IS PRINT POSITION < HALF OF DATA COUNT ?			
				3645	*	TAKE ONE-HALF OF COUNT ROUTINE (DIVIDE)			
				3646	*				
				3647	*****				
4E76	7C	00	E4	3648	MVI	DLFORK-1(,@BR),@ZERO			
4E79	4C	00	E5 03E3	3649	MVC	DLFORK(1,@BR),\$BUFPT	MOVE COUNT TO WORK AREA		
4E7E	5E	01	E5 E5	3650	ALC	DLFORK(2,@BR),DLFORK(,@BR)	ADD THREE TIMES		
4E82	5E	01	E5 E5	3651	ALC	DLFORK(2,@BR),DLFORK(,@BR)			
4E86	5E	01	E5 E5	3652	ALC	DLFORK(2,@BR),DLFORK(,@BR)			
4E8A	58	01	E4 E4	3653	MZN	DLFORK-1(,@BR),DLFORK-1(,@BR)	MOVE ZONE NUM		
4E8E	58	02	E4 E5	3654	MNZ	DLFORK-1(,@BR),DLFORK(,@BR)	DLFORK-1=1/2 NEXT LINE CNT		
				3655	*				
				3656	*	MOVE CARRAGE TO LEFT MARGIN OR TAB			
				3657	*				
4E92	8F	00	E4 03C1	3658	SLC	DFPPOS(1,@XR),\$LMRGN	PRPOS WITH IN WIDTH		
4E97	9D	00	E4 E4	3659	CLC	DFPPOS(1,@XR),DLFORK-1(,@BR)	IS PRPOS > 1/2 NEXT LINE		
4E9B	F2	82	2E	3660	JL	DLF900	SET TO GO TO LEFT MARGIN		
				3662	*****				
				3663	* DETERMINE TAB DIRECTION				
				3664	*****				
4E9E	1E	00	03E3 E7	3665	ALC	\$BUFPT(1),DLF001(,@BR)	COUNT PLUS ONE		
4EA3	0C	00	03E5 03C1	3666	MVC	\$LPROS(1),\$LMRGN	SET POSITION TO LEFT MARGIN		
4EA9	7C	01	9E	3667	MVI	DFP260-DFPRNT+2(,@BR),@B1	SET TO PRINT LEFT		
4EAC	8D	00	E4 03E3	3668	CLC	DFPPOS(1,@XR),\$BUFPT	COMPARE PRINT POS. TO LINE LNG		
4EB1	F2	81	23	3669	JE	DLF950	JUMP EQUAL LINE & POSITION		
4EB4	F2	84	10	3670	JH	DLF800	JUMP TO TAB LEFT		

DLPRT - LINE PRINTER ROUTINE

				3672	*		
				3673	*	COMPUTE TAB RIGHT	
				3674	*		
4EB7	2F	00	03E3 E4	3675	SLC	\$BUFPT(1),DFPPOS(,@XR)	GET TAB DISTANCE
4EBC	8C	00	E4 03E3	3676	MVC	DFPPOS(1,@XR),\$BUFPT	SAVE BUFFER POINTER
4EC1	7C	A0	DE	3677	MVI	DLFPCF(,@BR),DLTABR	SET TAB RIGHT OP
4EC4	F2	87	08	3678	J	DLF920	JUMP TO SET TAB COUNT

## DLFPRT - LINE PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 265
					3680	*				
					3681	*	COMPUTE LEFT TAB			
					3682	*				
				4EC7	3683	DLF800	EQU *			FIND TAB LEFT COUNT
4EC7	8F	00	E4 03E3		3684		SLC DFPPOS(1,@XR), \$BUFPT			GET TAB DISTANCE
				4ECC	3685	DLF900	EQU *			SET TAB LEFT
4ECC	7C	90	DE		3686		MVI DLFPCF(, @BR), DLTABL			SET TAB LEFT OP
				4ECF	3687	DLF920	EQU *			HARDWARE REQUIREMENT
4ECF	9F	00	E4 E7		3688		SLC DFPPOS(1,@XR), DLF001(, @BR)			ONE LESS
4ED3	6C	00	DF E4		3689		MVC DLFPCF+1(, @BR), DFPPOS(, @XR)			SET TAB COUNT
				4ED7	3690	DLF950	EQU *			SET AT LEFT MARGIN INDICATION
4ED7	2C	01	03EA DF		3691		MVC \$LPRIO, DLFBPT(2, @XR)			SAVE PDAR ADDR & BUFR. INCR.
4EDC	74	02	E5		3692		ST DLFOR(, @BR), @XR			SAVE XR
4EDF	B5	02	D9		3693		L BUFADR(, @XR), @XR			XR = CADDR LINE PRINTER BUFFER
4EE2	74	02	DD		3694		ST DFPAPC(, @BR), @XR			SAVE BUFFER ADDR
4EE5	7C	FB	DD		3695		MVI DFPAPC(, @BR), DLFCAR			GET DISP. TO COMMANDS
4EE8	9C	04	FF E2		3696		MVC BFPCRO-LPBUFR(5, @XR), DFPPCO(, @BR)			MOVE COMMANDS TO PCAR
4EEC	75	02	E5		3697		L DLFOR(, @BR), @XR			RESTORE XR TO VLPRT2
4EEF	3C	00	03E3		3698		MVI \$BUFPT, @ZERO			SET BUFFER PTR = 0
4EF3	D0	87	99		3699		B DFP250(, @BR)			GO TO DFPRNT TO DO I/O
					3700	*				
					3701	*****				
					3702	*****	END V\$LPR2		*****	
					3703	*****				

DLPRT - LINE PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 266
					3705	*****	*****			
					3706	*	LINE DRINTER BUFFER AREA			
					3707	*****	*****			
4F00					3708	ORG	*,256,0			
				4F00	3709	USING	LPBUFR,@XR			SET BASE FOR BUFFER AREA
				4F00	3710	LPBUFR EQU	*			LINE PRINTER BUFFER AREA
4F00				4FFA	3711	DS	CL251			LINE PRINTER BUFFER AREA
					3713	*****	LINE PRINTER COMMANDS PCAR			*****
				4FFB	3714	BFPCAR EQU	*			LINE PRINTER COMMANDS
4FFB	0000000000			4FFF	3715	DC	XL5'00'			LINE PRINTER COMMANDS
				4FFF	3716	BFPCRO EQU	*-1			LAST BYTE OF COMMANDS
				00FB	3717	DLFCAR EQU	BFPCAR-LPBUFR			DISPLACEMENT TO PCAR
					3718	*****	*****			

## VLPRT3 - BI-DIRECTIONAL PRINT ROUTINE CORRECTION PAGE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 267
5300				3720	ORG	X'5300'	PATCH AREA	1-5
				3721	*****			
				3722	*			*
				3723	*	THIS PAGE 15 USED BY THE BI-DIRECTIONAL PRINT ROUTINES TO CORRECT		*
				3724	*	PROBLEMS CONNECTED WITS APAR NUMBERS 968 AND 972. THE ROUTINES		*
				3725	*	USING THIS PAGE AND THEIR ENTRY POINTS ARE:		*
				3726	*	DFPRNT - VLPRT3, DFPENT		*
				3727	*	FZSPRT - VLPRT4		*
				3728	*	DLFPRT - VLPRT5, VLPRT6		*
				3729	*			*
				3730	*****			
				5300 3731	VLPRT3	EQU	*	DFPRNT INTERFACE 1-5
				5300 3732	DFPCHK	EQU	*	1-5
				2800 3733			USING DFPASE,@BR	1-5
				4D00 3734			USING DLFPRT,@XR	1-5
5300	7D	00	F6	3735	CLI	DFPIST+@PRCNT(,@BR),@ZERO	ANOTHER LINE TO PRINT	1-5
5303	F2	01	0B	3736	JNE	DFPENT	CONTINUE PROCESSING LINE	1-5
5306	F2	87	30	3737	J	DFPULK	GO TO UNLOCK ROUTINE	1-5
5309	C0	87	1354	3738	B	I\$LOCK	LOCK PAGE VLPRT3	1-5
530D	6C	03	F8 03	3739	MVC	DFPIST+@PLNGH-1(@PLNGH,@BR),@PLNGH-1(,@XR)	MOVE THE PRT	1-5
				3740	*		* PARAMETER LIST TO WRK AREA	1-5
5311	5C	02	F4 F8	3741	DFPENT	MVC	DFPDSV(@CADDR+1,@BR),DFPIST+@PDATA(,@BR)	MOVE THE PRT 1-5
				3742	*		* CNT AND DATA ADDRESS	1-5
5315	4C	00	FB 03C2	3743	MVC	DFPSYC+@SYCNT(1,@BR),\$PRPOS	SAVE HD POSITION FOR SYNC	1-5
531A	5C	01	DF F6	3744	MVC	DFPPCF+@PRCNT(2,@BR),DFPIST+@PRCNT(,@BR)	SET CTRL+CNT	1-5
531E	39	1E	03E4	3745	TBF	\$LPRP3,@KENAB	TEST FOP MATRIX PRINT MODE	1-5
5322	D0	90	23	3746	BF	DFP115(,@BR)	BRANCH IF MATRIX PRINT	1-5
5325	38	80	03D2	3747	TBN	\$IOIND,\$LNPTR	IS LINE PRINTER REQUESTED ?	1-5
5329	D0	90	23	3748	BF	DFP115(,@BR)	BRANCH IF NOT	1-5
532C	C0	87	1330	3749	B	I\$LDXR	BRANCH TO LOAD PAGE	1-5
5330	4D00			5331 3750	DC	AL(@VADDR)(V\$LPRT)	LINE PRINTER PAGE	1-5
5332	C0	87	1354	3751	B	I\$LOCK	GO LOCK PAGE	1-5
5336	E0	87	00	3752	B	@ZERO(,@XR)	BRANCH TO LINE PRINTER LINK	1-5
				3753	*			1-5
				5339 3754	DFPULK	EQU	*	UNLOCK ALL LINE PRINTER 1-5
				3755	*		* ROUTINE PAGES	1-5
5339	7C	80	A3	3756	MVI	DFP280+@Q-DFPASE(,@BR),@NOP	SET ERP INDR OFF	1-5
533C	1C	01	144A 1F	3757	MVC	I\$VADR,DFP105(2,@BR)	DLFPRT VM ADDR	1-5
5341	C0	87	1350	3758	B	I\$UNLK	UNLOCK PAGE	1-5
5345	3C	4E	1449	3759	MVI	I\$VADR-1,DLFX4E	VLPRT2 VM ADDR	1-5
5349	C0	87	1350	3760	B	I\$UNLK	UNLOCK PAGE	1-5
534D	3C	53	1449	3761	MVI	I\$VADR-1,DLFX53	VLPRT3 VM ADDR	1-5
5351	C0	87	1350	3762	B	I\$UNLK	UNLOCK PAGE	1-5
5355	C0	87	12D3	3763	B	I\$RTRN	BRANCH TO CALLING PGM-FZPRNT	1-5
				3764	*			1-5
				5359 3765	VLPRT4	EQU	*	FZSPRT INTERFACE 1-5
				3600 3766			USING FZSP3B,@BR	1-5
5359	4E	00	DB 03C2	3767	FZS991	ALC	FZS3CC(,@BR),\$PRPOS(1)	ADD PRT ZONE LNG TO CURRENT 1-5
535E	5D	00	DB 6A	3768	CLC	FZS3CC(,@BR),FZS3RM(1,@BR)	* CARRIER POSITION - BRANCH	1-5
5362	F2	84	03	3769	JH	FZS992	* IF RIGHT MGN IS EXCEEDED	1-5
5365	D0	87	B9	3770	B	FZS710(,@BR)	BRANCH BACK IF NOT	1-5
5368	38	80	03D2	3772	FZS992	TBN	\$IOIND,\$LNPTR	IS LINE PRINTER REQUESTED ? 1-5
536C	F2	90	03	3773	JF	FZS993	NO, DON'T SET CARRIAGE RTN	1-5
536F	7C	C0	F2	3774	MVI	FZS3PF(,@BR),@PRETR	SET CARRIAGE RETURN INDR	1-5
5372	D2	02	F2	3775	FZS993	LA	FZS3PL(,@BR),@XR	LOAD DATA OUTDUT PPL CADDR 1-5

## VLPRT3 - BI-DIRECTIONAL PRINT ROUTINE CORRECTION PAGE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 268
5375	C0	87	12B1		3776	B	I\$CALL			LINK TO EXECUTE PRINTER IOCR 1-5
5379	2800			537A	3777	DC	AL(@VADDR)(V\$SPRT)			MATRIX PRINTER IOCR VADDR 1-5
537B	7C	40	F2		3778	MVI	FZS3PF(, @BR), @PRINT			SET INDR TO PRINT ONLY 1-5
537E	0D	00	044A	0D5A	3779	CLC	\$PRDEV-1, I\$WRK2-1(1)			IF CRT IS NOT A SYSTEM PRINT 1-5
5384	F2	82	06		3780	JL	FZS994			* DEVICE, EXIT ROUTINE 1-5
5387	C0	87	12B1		3781	B	I\$CALL			LINK TO EXCUTE PRINT ON CRT 1-5
538B	3700			538C	3782	DC	AL(@VADDR)(FZS800)			PRINT CRT RTN VADDR 1-5
538D	C0	87	12D3		3783	FZS994 B	I\$RTRN			RETURN TO 1ST PRINT RTN PAGE 1-5
				5391	3785	VLPRT5 EQU	*			DLFPRT INTERFACE NO. 1 1-5
				2800	3786	USING	DFPASE, @BR			1-5
				4D00	3787	USING	DLFPRT, @XR			1-5
5391	5F	01	F2	E7	3788	SLC	DLFDSV-2(2, @BR), DLF001(, @BR)			COUNT LESS ONE 1-5
5395	BD	01	F0		3789	CLI	DLFSWC(, @XR), DLFRTN			IS SWITCH SET FOR RTN CARRAGE 1-5
5398	F2	81	04		3790	JE	DLF960			YES, DO NOT INCR DATA PTR 1-5
539B	5E	01	F8	F2	3791	ALC	DLFIST+@PDATA(2, @BR), DLFDSV-2(, @BR)			GET DATA ADDR PTR 1-5
539F	9C	01	62	F8	3792	DLF960 MVC	DLF150+@DOP2(2, @XR), DLFIST+@PDATA(, @BR)			SET DATA ADDR 1-5
53A3	9C	00	5F	F2	3793	MVC	DLF150+@VQ(1, @XR), DLFDSV-2(, @BR)			GET COUNT FOR MVC 1-5
53A7	8C	00	60	03E3	3794	MVC	DLF150+@D1(1, @XR), \$BUFPT			MOVE BUFFER DISP. INTO INST. 1-5
53AC	9F	00	60	E7	3795	SLC	DLF150+@D1(1, @XR), DLF001(, @BR)			DISP. LESS ONE 1-5
53B0	BC	00	F0		3796	MVI	DLFSWC(, @XR), X'00'			SET CARRAGE RETURN SW OFF 1-5
53B3	E0	87	5B		3797	B	DLF146(, @XR)			CONTINUE 1-5
					3798	*				
				53B6	3799	VLPRT6 EQU	*			DLFPRT INTERFACE NO. 2 1-5
53B6	7C	40	F5		3800	MVI	DLFIST+@PCTRL(, @BR), @PRINT			SET PRINT ONLY 1-5
53B9	6C	00	F6	DC	3801	MVC	DLFIST+@PRCNT(, @BR), DFPRES(1, @XR)			BUF PTR - RESIDUAL 1-5
53BD	6C	00	F2	DC	3802	MVC	DLFDSV-2(, @BR), DFPRES(1, @XR)			DATA COUNT - RESIDUAL 1-5
53C1	0C	00	03C2	03C1	3803	MVC	\$PRPOS(1), \$LMRGN			SET DUMMY POSITION-LEFT MGN. 1-5
53C7	BC	01	F0		3804	MVI	DLFSWC(, @XR), DLFRTN			SET SWITCH FOR RTN CARRIAGE 1-5
53CA	E0	87	25		3805	B	DLF100(, @XR)			CONTINUE PROCESSING 1-5
					3806	*##### X'5400' #####				
					3807	* N O T S C A N N E D (GENERAL PURPOSE BUFFERS 1 & 2.)				
					3808	*##### X'55FF' #####				
				FFFF	3809	END				

TOTAL STATEMENTS IN ERROR IN THIS ASSEMBLY = 0

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 269

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$\$\$CMD	001	0020	0660	
\$\$\$DAT	001	0040	0659	
\$\$\$EPL	001	0091	0656	
\$\$\$ERN	001	0080	0710	
\$\$\$FUN	001	0010	0661	
\$\$\$NLN	001	00A0	0706	
\$\$\$STD	001	0081	0655	
\$\$\$001	040	2DF8	0406	
\$\$BNLN	001	0605	0636	0638
\$\$CDBS	001	08C0	0686	
\$\$CDND	001	0666	0645	
\$\$CDRD	001	0890	0684	0686
\$\$CKEY	001	0603	0634	
\$\$CKFF	001	0B3D	0666	
\$\$COFF	001	0B44	0665	
\$\$CSNS	001	209C	0695	
\$\$DATB	001	0BBF	0667	
\$\$EOSA	001	0AFE	0664	
\$\$ERSK	001	1C00	0705	
\$\$FITS	001	1D00	0713	
\$\$FLIB	001	06FF	0712	
\$\$ILEN	001	0601	0630	0632 0636
\$\$ILHD	001	0600	0628	0630
\$\$INLN	001	0607	0643	0645 0647
\$\$INND	001	06FA	0647	
\$\$KBDT	001	09E1	0654	0658
\$\$KBSN	001	09E2	0658	0663
\$\$KLD1	001	0600	0718	7878 8609
\$\$KLD2	001	0700	0720	
\$\$KLD3	001	0C00	0722	
\$\$LPOS	001	09EB	0663	
\$\$PCNT	001	07E9	0679	
\$\$PLYN	001	2004	0693	7677 9167
\$\$PRES	001	0890	0652	0654 0664 0665 0666 0667 0684
\$\$PRFL	001	2143	0697	
\$\$PRNT	001	0707	0673	0674 0678 0679 7676
\$\$PRTN	001	0782	0674	
\$\$PSIO	001	07CE	0678	
\$\$PYCD	001	2200	0699	
\$\$PYMP	001	2000	0691	0693 0695 0697 0699
\$\$SLIB	001	1C00	0708	
\$\$TPCD	001	0606	0638	0643
\$\$UPAR	001	0602	0632	0634
\$\$WSPB	001	1E00	0711	
\$\$XIND	001	06FF	0709	0712
\$\$ZERO	001	0000	0224	0225 0227 0228 0229 0233 0691
\$ABORT	001	0010	0337	
\$BASIC	001	0080	0395	
\$BIGCD	001	0080	0471	7946
\$BLDPL	001	0579	0604	0606
\$BLNOE	001	0569	0594	
\$BLOAD	001	0522	0585	0587 0590 0603 0604 7826 8475
\$BLRTN	001	0550	0593	0594
\$BRSAV	001	03C5	0282	0283 8916* 8936
\$BSADR	001	0587	0609	0611
\$BUFPT	001	03E3	0490	0491 8731 9519* 9533* 3468* 3495 3598 3614* 3620* 3624 3634 3636*



CROSS REFERENCE																
SYMBOL	LEN	VALUE	DEFN	REFERENCES									VER 15,	MOD 00	31/05/21	PAGE 270
\$CABLD	001	04B4	0563	3637 0564	3649	3665*	3668	3675*	3676	3684	3698*	3794				
\$CAERK	001	0469	0540	0543												
\$CAERR	001	03CD	0288	0290												
\$CAIPL	001	049D	0559	0561												
\$CALLI	001	0008	0480													
\$CARDI	001	0001	0251													
\$CARPL	001	04A1	0561	0563												
\$CIENT	001	0483	0550	0551	9043	9052										
\$CIEXT	001	0480	0549	0550												
\$CIMSK	001	0476	0546	0549												
\$CISUS	001	0496	0554	0559	0005											
\$CLBFR	001	0010	0438													
\$CMDKY	001	0008	0350													
\$CMODE	001	0002	0400													
\$CONFIG	001	03DD	0463	0473	7946											
\$CRPOS	001	03E2	0489	0490	2809	2816	2923									
\$CRTAD	001	044D	0528	0529												
\$CRTAV	001	0002	0344													
\$CRTDN	001	0002	0368													
\$CRTIN	001	03D3	0365	0372												
\$CRTNO	001	0004	0347													
\$CRTPU	001	0004	0369													
\$CRTSP	001	0008	0370													
\$CRTUP	001	0001	0367													
\$CRUSH	001	0080	0476													
\$CSDPL	001	050E	0575	0576												
\$C0001	001	0464	0532	0538												
\$DATE	001	043A	0513	0514												
\$DBGUF	001	03E0	0475	0484	7412											
\$DBLOK	001	0001	0425													
\$DFDET	001	03E8	0496	0497												
\$DISKN	001	0025	0227	7817	7837	8464	8489	3266	3277							
\$DKERR	001	0008	0406													
\$DKSIZ	001	03D7	0450	0458	0499											
\$DK100	001	0001	0452													
\$DK200	001	0002	0453													
\$DK400	001	0004	0454													
\$DK600	001	0008	0455													
\$DK800	001	0010	0456													
\$DPLSV	001	0449	0524	0526	7833	8483										
\$DTNMB	001	0040	0271													
\$DTRDR	001	0040	0359													
\$ENDNU	001	0600	0618	0628	0652	0673	0709	0718	0720	0722	2753					
\$ERDPL	001	046F	0543	0545												

VER 15, MOD 00 31/05/21 PAGE 271

SYMBOL	LEN	VALUE	DEFN	REFERENCES												VER 15, MOD 00 31/05/21 PAGE 271											
\$ER1N2	001	0050	0300																								
\$EXADR	001	0517	0578	0580																							
\$EXCMD	001	0001	0332																								
\$EXFTR	001	043B	0514	0519	7593	8924	0238	2180	3158	3161	3163																
\$FCIND	001	0010	0410																								
\$FDIND	001	0040	0417																								
\$FEARR	001	0004	0225																								
\$FEMAP	001	0588	0611	0612																							
\$FILIB	001	03DA	0461	0462																							
\$FITIN	001	0010	0386																								
\$FUIND	001	0020	0415																								
\$GUFIO	001	0583	0608	0609																							
\$GUFIR	001	0008	0260																								
\$HISTE	001	042E	0511	0512	9478*	9528*																					
\$HIST1	001	0435	0512	0513	9293*	9473*																					
\$HRDER	001	0020	0356	9290	9527																						
\$INDR1	001	03D4	0372	0398																							
\$INDR2	001	03D5	0398	0423	9298*	9471	9474*																				
\$INDR3	001	03D6	0423	0450																							
\$INLNO	001	03CF	0290	0292	0304	0311	0143																				
\$INRPT	001	0020	0268																								
\$IOIND	001	03D2	0339	0365	9290*	9527*	3747	3772																			
\$IOPGS	001	0010	0479	7412																							
\$IOYES	001	0002	0254																								
\$IPLDV	001	05FF	0615	0618																							
\$IRKEY	001	0020	0478																								
\$KEYBD	001	03E1	0484	0489																							
\$KEYCD	001	03C3	0248	0282																							
\$KEYDT	001	0040	0392																								
\$KE090	001	00DE	0228																								
\$KE130	001	01D5	0229																								
\$KYBSY	001	0010	0265																								
\$LDRTN	001	0571	0603																								
\$LEVEL	001	03DF	0473	0475																							
\$LIST	001	0002	0427																								
\$LMRGN	001	03C1	0243	0245	8929	9359	9362	9490	3528	3600	3633	3638	3658	3666	3803												
\$LNPTR	001	0080	0362	3747	3772																						
\$LOADB	001	054A	0587																								
\$LOADR	001	051A	0580	0583																							
\$LPRIO	001	03EA	0497	9518	3691*																						
\$LPROS	001	03E5	0492	0494	8741	9467	9508*	3463*	3632	3633*	3634*	3666*															
\$LPRP3	001	03E4	0491	0492	8630*	8707*	8739	8742*	9172*	9175*	9465	9468*	9485	9505	9507*												
				9529	9994*	0058*	0259*	3435*	3460	3462*	3534*	3745															
\$MOUNT	001	0020	0441																								
\$MPDWN	001	0001	0341	9527																							
\$NEXTB	001	03E6	0494	0495																							
\$NEXTL	001	03E7	0495	0496																							
\$NOENB	001	0008	0433																								
\$NOLST	001	0004	0257																								
\$NUCBS	001	03C0	0240	0241																							
\$NWRKF	001	0080	0446																								
\$NWRKR	001	0040	0443																								
\$PASWD	001	042D	0510	0511																							
\$PAUSD	001	04BA	0564	0566																							
\$PAUSE	001	0002	0334																								
\$PGMDT	001	0020	0389																								

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 272

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$PGMST	001	0010	0353	
\$PKERT	001	0419	0508	0510
\$PLST1	001	0454	0529	0530
\$PLST2	001	045B	0530	0531
\$PLST3	001	0462	0531	0532
\$PRDEV	001	044B	0526	0528 7585 7588* 7592* 7593* 7608* 9165 9170 0242 0254 2451 2625 3779
\$PRESN	001	0002	0377	
\$PROCI	001	0001	0374	
\$PRPOS	001	03C2	0245	0248 8741* 9333* 9339 9350* 9357 9362* 9467* 9501* 9508 2526 2533 2649 3463 3469* 3528* 3743 3767 3803*
\$PSDBR	001	04FA	0569	
\$PSDXR	001	04F2	0568	0569
\$PSTEP	001	0004	0335	
\$PSTMT	001	0008	0336	
\$PTCH1	001	03F5	0499	0503
\$READY	001	0080	0419	
\$REORD	001	0040	0477	
\$RLOAD	001	051E	0583	0585
\$RMRGN	001	03C0	0241	0243 8928 9340 2456 3599
\$RSTR	001	04D6	0566	0568 0570 0575
\$RUNIT	001	0001	0313	
\$SFAID	001	050D	0571	
\$SPRNT	001	0465	0538	0540
\$SRTRN	001	04FE	0570	0571
\$STEPT	001	0002	0314	
\$SWPCR	001	0511	0576	0578
\$TABLN	001	03CB	0285	0288
\$TFLOW	001	0008	0320	
\$TRACE	001	0004	0315	
\$TRALL	001	0010	0321	
\$TROVR	001	054E	0590	0593
\$TRUNK	001	0080	0273	
\$TRVAR	001	0020	0322	
\$UNMSK	001	048D	0551	0554
\$USRDR	001	03DC	0462	0463
\$VMDEF	001	0080	0326	
\$VOLF1	001	03FE	0505	0506
\$VOLF2	001	040E	0507	
\$VOLID	001	03F6	0503	0504 0508
\$VOLR1	001	03F6	0504	0505
\$VOLR2	001	0406	0506	0507
\$WAITF	001	057F	0606	0608 7838 8490 3278
\$WFDEF	001	0040	0520	
\$WFLOK	001	0008	0383	
\$WFNME	001	0443	0519	0524
\$WSIND	001	0004	0380	
\$XIND1	001	03D0	0311	0330
\$XIND2	001	03D1	0330	0339
\$XIND3	001	03D8	0458	0461
\$XPREC	001	0040	0323	
\$XRSAB	001	03C7	0283	0285
\$ZTRAD	001	05A2	0612	
\$12K	001	0004	0467	
\$16CKY	001	0008	0469	
\$16K	001	0002	0466	

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 31/05/21 PAGE 273

\$22IMP 001 0001 0464  
###BL 001 0000 1437  
###CK 001 0000 1565  
###CN 001 0000 1533  
###CO 001 0000 1325  
###CS 001 0000 1385  
###DR 001 0000 1129  
###ER 001 0000 1329  
###FS 001 0000 1425  
###IN 001 0000 1569  
###PW 001 0000 1573  
###RS 001 0000 1405  
###SA 001 0000 1393  
###SS 001 0000 1389  
###VU 001 0600 1349  
###0T 001 0700 1121  
###1T 001 0000 1125  
###BCO 001 0600 1137  
###BOV 001 0800 1409  
###DPR 001 0700 1145  
###DRE 001 0889 1161  
###DSP 001 2800 1181  
###ECM 001 0C00 1441  
###EFK 001 0C00 1461  
###ERR 001 0C00 1433  
###EXM 001 0C00 1321  
###FIL 001 0E00 1401  
###FIS 001 0E00 1397  
###FML 001 0200 1529  
###FMS 001 0200 1369  
###GRA 001 0889 1293  
###GUF 001 0C00 1429  
###INL 001 0600 1509  
###INS 001 0600 1133  
###KAL 001 0C00 1297  
###KCA 001 0C00 1513  
###KCH 001 0C00 1265  
###KCN 001 0C00 1381  
###KCT 001 0C00 1233  
###KDE 001 0C00 1229  
###KDI 001 0D00 1309  
###KDN 001 0C00 1217  
###KDO 001 0E00 1313  
###KED 001 0C00 1153  
###KEN 001 0C00 1157  
###KEX 001 0C00 1177  
###KGO 001 0C00 1149  
###KHE 001 0C00 1333  
###KKE 001 0C00 1561  
###KLI 001 0C00 1237  
###KLL 001 0920 1537  
###KLO 001 0C00 1241  
###KME 001 0D00 1221  
###KMO 001 0C00 1165  
###KNA 001 0C00 1277  
###KOV 001 0E00 1197

3965

7858 8585

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 274

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$\$\$KPA	001	0C00	1173	
\$\$\$KPO	001	0C00	1261	
\$\$\$KPR	001	0C00	1285	
\$\$\$KRE	001	0C00	1205	
\$\$\$KRL	001	0700	1301	
\$\$\$KRM	001	0C00	1169	
\$\$\$KRN	001	0700	1189	
\$\$\$KRO	001	0D00	1193	
\$\$\$KRS	001	0C00	1517	
\$\$\$KRU	001	0C00	1213	
\$\$\$KRV	001	0800	1305	
\$\$\$KSA	001	0C00	1249	
\$\$\$KSE	001	0E00	1289	
\$\$\$KSO	001	0C20	1341	
\$\$\$KSS	001	0C00	1273	
\$\$\$KSV	001	0980	1269	
\$\$\$KSY	001	0C00	1281	
\$\$\$KWI	001	0C00	1209	
\$\$\$KWR	001	0C00	1201	
\$\$\$LOA	001	0600	1141	
\$\$\$MIP	001	0C00	1337	
\$\$\$SDS	001	0C00	1449	
\$\$\$SFF	001	0E00	1453	
\$\$\$SFL	001	0F00	1445	7868 8597
\$\$\$SFO	001	1500	1417	
\$\$\$SFS	001	0C00	1413	
\$\$\$SPA	001	0C00	1253	
\$\$\$SPO	001	0806	1257	
\$\$\$SPS	001	0C00	1245	
\$\$\$STR	001	1600	1421	
\$\$\$TDC	001	1000	1225	
\$\$\$TSY	001	1000	1185	
\$\$\$TVK	001	0FC0	1361	
\$\$\$UAL	001	0C00	1377	
\$\$\$UAT	001	0900	1473	
\$\$\$UCD	001	0900	1481	
\$\$\$UCN	001	0C00	1465	
\$\$\$UCP	001	0700	1469	
\$\$\$UDE	001	0C00	1485	
\$\$\$UDI	001	0C00	1489	
\$\$\$UEX	001	0C00	1373	
\$\$\$UIN	001	0C00	1477	
\$\$\$UPA	001	0C00	1457	
\$\$\$UPO	001	0C00	1525	
\$\$\$UPT	001	0C00	1521	
\$\$\$VCR	001	2000	1317	
\$\$\$VLO	001	0600	1353	
\$\$\$VOD	001	0600	1357	
\$\$\$VVM	001	0000	1365	
\$\$\$VXI	001	0600	1345	
\$\$\$ZDU	001	1100	1497	
\$\$\$ZLB	001	1100	1541	
\$\$\$ZLO	001	1100	1501	
\$\$\$ZLV	001	0F00	1557	
\$\$\$ZL1	001	0F00	1545	
\$\$\$ZL2	001	0F00	1549	

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 31/05/21 PAGE 275

\$\$\$ZL3	001	0C00	1553	
\$\$\$ZTR	001	1000	1493	
\$\$\$ZUT	001	0C00	1505	
\$\$#BLN	001	18D4	1436	
\$\$#CKT	001	2118	1564	
\$\$#CNF	001	2000	1532	
\$\$#COR	001	0800	1324	
\$\$#CSA	001	1000	1384	
\$\$#DRT	001	0000	1128	
\$\$#ERM	001	0928	1328	
\$\$#FSP	001	1880	1424	
\$\$#INV	001	212C	1568	
\$\$#PWR	001	2300	1572	
\$\$#RSP	001	1780	1404	
\$\$#SAV	001	1180	1392	
\$\$#SSA	001	1128	1388	
\$\$#VUF	001	0B08	1348	
\$\$#0TR	001	0000	1120	
\$\$#1TR	001	0080	1124	
\$\$@#BL	001	0001	1438	
\$\$@#CK	001	0004	1566	
\$\$@#CN	001	0001	1534	
\$\$@#CO	001	003A	1326	
\$\$@#CS	001	003A	1386	
\$\$@#DR	001	0008	1130	
\$\$@#ER	001	0032	1330	
\$\$@#FS	001	0030	1426	
\$\$@#IN	001	003A	1570	
\$\$@#PW	001	00C0	1574	
\$\$@#RS	001	0030	1406	
\$\$@#SA	001	0108	1394	
\$\$@#SS	001	0001	1390	
\$\$@#VU	001	0002	1350	
\$\$@#0T	001	0018	1122	
\$\$@#1T	001	0018	1126	
\$\$@BCO	001	0018	1138	
\$\$@BOV	001	0018	1410	
\$\$@DPR	001	0005	1146	
\$\$@DRE	001	0001	1162	
\$\$@DSP	001	0004	1182	
\$\$@ECM	001	0006	1442	
\$\$@EFK	001	0002	1462	
\$\$@ERR	001	0003	1434	
\$\$@EXM	001	0003	1322	
\$\$@FIL	001	0009	1402	
\$\$@FIS	001	0009	1398	
\$\$@FML	001	0052	1530	
\$\$@FMS	001	0052	1370	
\$\$@GRA	001	0003	1294	
\$\$@GUF	001	0010	1430	
\$\$@INL	001	0010	1510	
\$\$@INS	001	0010	1134	
\$\$@KAL	001	000F	1298	
\$\$@KCA	001	000C	1514	
\$\$@KCH	001	000C	1266	
\$\$@KCN	001	0010	1382	

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 31/05/21 PAGE 276

##\$@KCT	001	0009	1234
##\$@KDE	001	0010	1230
##\$@KDI	001	0005	1310
##\$@KDN	001	0010	1218
##\$@KDO	001	000C	1314
##\$@KED	001	000E	1154
##\$@KEN	001	0006	1158
##\$@KEX	001	0003	1178
##\$@KGO	001	0002	1150
##\$@KHE	001	000C	1334
##\$@KKE	001	0006	1562
##\$@KLI	001	0011	1238
##\$@KLL	001	0001	1538
##\$@KLO	001	0008	1242
##\$@KME	001	0003	1222
##\$@KMO	001	0004	1166
##\$@KNA	001	0008	1278
##\$@KOV	001	0009	1198
##\$@KPA	001	0005	1174
##\$@KPO	001	000D	1262
##\$@KPR	001	0009	1286
##\$@KRE	001	0002	1206
##\$@KRL	001	0004	1302
##\$@KRM	001	0003	1170
##\$@KRN	001	0003	1190
##\$@KRO	001	000A	1194
##\$@KRS	001	000A	1518
##\$@KRU	001	0003	1214
##\$@KRV	001	000D	1306
##\$@KSA	001	0011	1250
##\$@KSE	001	0004	1290
##\$@KSO	001	000D	1342
##\$@KSS	001	000B	1274
##\$@KSV	001	0002	1270
##\$@KSY	001	000F	1282
##\$@KWI	001	0002	1210
##\$@KWR	001	0002	1202
##\$@LOA	001	0013	1142
##\$@MIP	001	000D	1338
##\$@SDS	001	0004	1450
##\$@SFF	001	0008	1454
##\$@SFL	001	0005	1446
##\$@SFO	001	0003	1418
##\$@SFS	001	0011	1414
##\$@SPA	001	0004	1254
##\$@SPO	001	0003	1258
##\$@SPS	001	0001	1246
##\$@STR	001	0002	1422
##\$@TDC	001	0003	1226
##\$@TSY	001	0003	1186
##\$@TVK	001	0001	1362
##\$@UAL	001	0011	1378
##\$@UAT	001	000C	1474
##\$@UCD	001	000B	1482
##\$@UCN	001	0009	1466
##\$@UCP	001	000F	1470



## CROSS REFERENCE

SYMBOL   LEN   VALUE   DEFN   REFERENCES   VER 15, MOD 00   31/05/21   PAGE 277

#\$@UDE	001	000E	1486	
#\$@UDI	001	0008	1490	
#\$@UEX	001	000E	1374	
#\$@UIN	001	000F	1478	
#\$@UPA	001	0004	1458	
#\$@UPO	001	0005	1526	
#\$@UPT	001	0012	1522	
#\$@VCR	001	0008	1318	
#\$@VLO	001	0002	1354	
#\$@VOD	001	0016	1358	
#\$@VVM	001	0030	1366	
#\$@VXI	001	0002	1346	
#\$@ZDU	001	0008	1498	
#\$@ZLB	001	0002	1542	
#\$@ZLO	001	000C	1502	
#\$@ZLV	001	0006	1558	
#\$@ZL1	001	0007	1546	
#\$@ZL2	001	000D	1550	
#\$@ZL3	001	000A	1554	
#\$@ZTR	001	0001	1494	
#\$@ZUT	001	0014	1506	
#\$BCOM	001	0080	1136	
#\$BOLV	001	1780	1408	
#\$DPRI	001	014C	1144	
#\$DREA	001	0200	1160	
#\$DSPL	001	0240	1180	
#\$ECMA	001	1900	1440	
#\$EFKE	001	1990	1460	
#\$ERRP	001	18C0	1432	
#\$EXMS	001	07D4	1320	
#\$FILN	001	1724	1400	
#\$FIST	001	1700	1396	
#\$FMLN	001	1E00	1528	
#\$FMST	001	0D00	1368	
#\$GRAP	001	0690	1292	
#\$GUFU	001	1880	1428	
#\$INLN	001	1C84	1508	
#\$INST	001	0020	1132	
#\$KALL	001	06A4	1296	
#\$KCAL	001	1CC4	1512	
#\$KCHA	001	053C	1264	
#\$KCND	001	0F80	1380	
#\$KCTL	001	03BC	1232	
#\$KDEL	001	035C	1228	
#\$KDIS	001	0744	1308	
#\$KDNT	001	0300	1216	
#\$KDOV	001	0780	1312	
#\$KEDI	001	0188	1152	
#\$KENA	001	01C4	1156	
#\$KEXT	001	0234	1176	
#\$KGOS	001	0180	1148	
#\$KHEL	001	0A30	1332	
#\$KKEY	001	2100	1560	
#\$KLIS	001	0400	1236	
#\$KLLA	001	2004	1536	
#\$KLOG	001	0444	1240	

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 31/05/21 PAGE 278

#\$KMER	001	030C	1220
#\$KMOU	001	0204	1164
#\$KNAM	001	05C0	1276
#\$KOVN	001	0290	1196
#\$KPAS	001	0220	1172
#\$KPOO	001	0508	1260
#\$KPRT	001	063C	1284
#\$KREA	001	02BC	1204
#\$KRLA	001	0700	1300
#\$KRMO	001	0214	1168
#\$KRNU	001	0280	1188
#\$KROV	001	028C	1192
#\$KRSU	001	1D24	1516
#\$KRUN	001	02CC	1212
#\$KRVL	001	0710	1304
#\$KSAV	001	0488	1248
#\$KSET	001	0680	1288
#\$KSOV	001	0AC8	1340
#\$KSSP	001	0594	1272
#\$KSVL	001	058C	1268
#\$KSYM	001	0600	1280
#\$KWID	001	02C4	1208
#\$KWRI	001	02B4	1200
#\$LOAD	001	0100	1140
#\$MIPP	001	0A80	1336
#\$SDSY	001	192C	1448
#\$SFFI	001	193C	1452
#\$SFLO	001	1918	1444
#\$SFOV	001	1844	1416
#\$SFSY	001	1800	1412
#\$SPAC	001	04CC	1252
#\$SPOV	001	04DC	1256
#\$SPSY	001	0484	1244
#\$STRO	001	1850	1420
#\$TDCK	001	0350	1224
#\$TSYK	001	0250	1184
#\$TVKB	001	0BAC	1360
#\$UALL	001	0F00	1376
#\$UATR	001	1A38	1472
#\$UCDI	001	1AD8	1480
#\$UCNF	001	19B8	1464
#\$UCPL	001	19DC	1468
#\$UDEL	001	1B24	1484
#\$UDIS	001	1B5C	1488
#\$UEXL	001	0EA8	1372
#\$UINI	001	1A88	1476
#\$UPAC	001	1980	1456
#\$UPOV	001	1D24	1524
#\$UPTF	001	1D5C	1520
#\$VCRT	001	07B4	1316
#\$VLOA	001	0B80	1352
#\$VODK	001	0B88	1356
#\$VVMR	001	0C00	1364
#\$VXIT	001	0B00	1344
#\$ZDUM	001	1BA4	1496
#\$ZLBM	001	2008	1540

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 279

SYMBOL	LEN	VALUE	DEFN	REFERENCES
#\$ZLOA	001	1BC4	1500	
#\$ZLVR	001	20B0	1556	
#\$ZL1M	001	2010	1544	
#\$ZL2M	001	2030	1548	
#\$ZL3M	001	2088	1552	
#\$ZTRA	001	1B9C	1492	
#\$ZUTM	001	1C14	1504	
#@#BAD	001	0455	0880	
#@#IO1	001	0459	0888	
#@#IO2	001	045D	0889	
#@#TAT	001	0941	0916	
#@#TBA	001	09A1	0920	
#@#TFS	001	0941	0914	
#@#TSY	001	0941	0918	
#@#VFP	001	0700	0906	
#@#VLP	001	093D	0909	
#@#WDB	001	050C	0901	
#@#WFT	001	0500	0899	
@@#BA	001	0001	0881	
@@#IO	001	0001	0893	
@@#SC	001	0002	0890	
@@#TA	001	0010	0917	
@@#TB	001	0010	0921	
@@#TS	001	0005	0919	
@@#TW	001	0020	0915	
@@#VM	001	0100	0910	
@@#WD	001	00BD	0902	
@@#WF	001	0003	0900	
@@#04	001	0004	0892	
@@#08	001	0008	0891	
@@BOV	001	0018	0869	
@@ECM	001	0006	0883	
@@ERR	001	0003	0877	
@@GUF	001	0010	0873	
@@LDS	001	0002	0879	
@@SDS	001	0004	0875	
@@SFF	001	0008	0887	
@@SFL	001	0005	0885	7867 8596
@@SFO	001	0005	0895	
@@SFS	001	0011	0871	
@@VSF	001	0010	0923	
@@VSL	001	000F	0924	7857 8584
@@VTR	001	0001	0908	
#@BOVL	001	0400	0868	
#@CORS	001	0005	0774	
#@ECMA	001	0481	0882	
#@ERRP	001	0441	0876	
#@GUFU	001	0401	0872	
#@LDSV	001	044D	0878	
#@MVSD	001	0001	0782	
#@NERO	001	0003	0776	
#@OBRA	001	0002	0778	
#@PTFL	001	0006	0797	
#@PTFS	001	0001	0796	
#@SDSY	001	04AD	0874	
#@SFFI	001	04BD	0886	

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 280

SYMBOL	LEN	VALUE	DEFN	REFERENCES
#@SFLO	001	0499	0884	7866 8595
#@SFOV	001	04C4	0894	
#@SFSY	001	0480	0870	
#@VCNT	001	0002	0794	
#@VLAB	001	0001	0789	
#@VLSD	001	0001	0780	
#@VSFI	001	09A1	0922	7856 8583
#@VTRL	001	0708	0907	
#@WAF1	001	0401	0867	
#@WAR1	001	0400	0866	
#CNDIS	001	0001	0749	
#CNFIG	001	0005	0785	
#CORSV	001	0010	0773	
#DKEXT	001	0002	0756	
#FIGSC	001	0001	0786	
#FMSTD	001	0000	0002	
#HISCT	001	0006	0763	
#HISDX	001	0003	0758	
#HISLN	001	0008	0755	0756 9293 9473
#HISN1	001	0003	0761	
#HISN2	001	0005	0762	
#HISTC	001	0007	0765	
#HISTN	001	0009	0767	
#HISTQ	001	0000	0759	
#HISTR	001	0001	0760	
#HISTS	001	0008	0766	
#HISTV	001	000F	0768	
#HSEND	001	0007	0764	
#HSENT	001	0001	0757	
#IOSDR	001	0019	0784	
#MVSDR	001	000D	0781	
#NEROV	001	009C	0775	
#OBRAD	001	001D	0777	
#PKCNT	001	0002	0742	
#PKMRW	001	002B	0743	
#PKRDD	001	0003	0740	
#PKRTD	001	0003	0739	
#PKRTL	001	0004	0746	
#PKVRD	001	000B	0744	
#PKVWD	001	0007	0745	
#PKWTD	001	0001	0741	
#PTFDA	001	00DC	0795	
#RDWTL	001	0004	0747	
#SDRDK	001	0011	0783	
#VLSDR	001	000C	0779	
#VLTBE	001	0008	0734	
#VOLF1	001	0009	0787	
#VOLNG	001	0006	0732	0734 0756
#VOLOC	001	0005	0733	
#VOLR1	001	0008	0788	
#VTCF1	001	0025	0791	
#VTCF2	001	0027	0793	
#VTCR1	001	0024	0790	
#VTCR2	001	0026	0792	
@\$D1BF	001	0008	2230	7401 7405
@\$D1DC	001	0000	2229	

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 281

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@SD1DF	001	001E	2234	
@SD1DP	001	0016	2233	
@SD1DV	001	000E	2232	
@SD1E1	001	0000	2223	
@SD1FS	001	000A	2231	
@SD1SW	001	001F	2236	7400
@SD2AS	001	0002	2241	
@SD2BS	001	0003	2248	7689 7797 8455
@SD2CB	001	0005	2251	7698 8010* 8011* 8012* 8161 8164* 8451 8498* 8502* 8529* 8542*
@SD2CF	001	0001	2240	7427* 7428 7572 8143 8650* 8652 8701*
@SD2CP	001	0005	2249	7694 7756* 7841 7887 8319 8357* 8372 8507 8538* 8679 8679*
@SD2CS	001	0004	2250	7689 7740 7797 8453 8455 8516* 8690
@SD2CY	001	0006	2252	
@SD2DA	001	0007	2253	
@SD2DC	001	0000	2245	7429 7586 7590 7618 8156 8656 8687
@SD2DD	001	0009	2254	7738 8680 8680*
@SD2EE	001	000F	2257	8690 8690*
@SD2E1	001	0040	2244	7446 7457 8650
@SD2FS	001	000B	2255	7737
@SD2IO	001	0001	2246	7573 7575 7577 7581* 7597 7627 7699 7701 7799 7890 7978 8144 8146 8148 8152* 8313 8496 8658 8660 8662* 8670* 8686* 8689 8689*
@SD2LC	001	000D	2256	8301 8317 8358* 8371 8499* 8544* 8681 8681*
@SD2PN	001	000A	2242	
@SD2SF	001	000B	2243	
@SD2VB	001	0002	2247	7695 7842 7888 8165 8320 8373 8508
@\$L1BF	001	0008	2263	
@\$L1DC	001	0001	2262	
@\$L1DF	001	0008	2265	7379 7405
@\$L1DP	001	0008	2266	
@\$L1DV	001	0006	2267	
@\$L1E	001	0020	2261	7410
@\$L1FS	001	0002	2264	
@\$L2AS	001	0001	2273	
@\$L2BS	001	0001	2280	
@\$L2CB	001	0001	2283	7698 7895 7945 7964 8010 8011 8012 8529 8542 8606 8607
@\$L2CF	001	0002	2272	
@\$L2CP	001	0002	2281	7694 7756 7841 7887 8319 8357 8372 8416 8538 8679
@\$L2CS	001	0001	2282	7740 8455 8516 8605
@\$L2DA	001	0002	2284	
@\$L2DC	001	0001	2277	8689
@\$L2DD	001	0002	2285	7738 8680
@\$L2E	001	0010	2276	7458 8690
@\$L2FS	001	0002	2286	7737 7739 7741
@\$L2HD	001	0040	2271	
@\$L2IO	001	0001	2278	7699 8689
@\$L2LC	001	0002	2287	8301 8317 8358 8371 8423 8499 8544 8604 8681
@\$L2PN	001	0008	2275	
@\$L2SF	001	0002	2274	
@\$L2VB	001	0001	2279	7689 7695 7797 7842 7888 7984 8165 8320 8373 8508
@\$MBCD	001	0020	2301	7618 8156
@\$MBCR	001	0008	2303	7590
@\$MBEN	001	000C	2291	8721
@\$MBND	001	0000	2298	
@\$MBPD	001	0080	2299	
@\$MBPT	001	0010	2302	7586
@\$MBPU	001	0001	2294	

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 31/05/21 PAGE 282

@\$MBSD	001	0040	2300	8687							
@\$M2CI	001	0008	2318	7575	8144	8152	8658	8686			
@\$M2CO	001	0004	2319	7573	7581	8146	8660	8686			
@\$M2EF	001	0002	2293	7597	7627	7701	7799	7890	7978	8662	8670
@\$M2FI	001	0080	2307	8148							
@\$M2FO	001	0040	2308	7577							
@\$M2FP	001	0020	2309	7712	8313						
@\$M2FT	001	0010	2312	8496							
@\$M2NS	001	00FF	2292								
@@E001	001	0000	2111	2113							
@@E003	001	0001	2113	2115							
@@E004	001	0002	2115	2117							
@@E005	001	0003	2117	2119							
@@E006	001	0004	2119	2121							
@@E007	001	0005	2121	2123							
@@E008	001	0006	2123	2125							
@@E009	001	0007	2125	2127							
@@E010	001	0008	2127	2129							
@@E011	001	0009	2129	2131							
@@E012	001	000A	2131	2133							
@@E013	001	000B	2133	2135							
@@E014	001	000C	2135	2137							
@@E015	001	000D	2137	2139							
@@E016	001	000E	2139	2141							
@@E017	001	000F	2141	2143							
@@E018	001	0010	2143	2145							
@@E019	001	0011	2145	2147							
@@E020	001	0012	2147	2149							
@@E021	001	0013	2149	2151							
@@E023	001	0014	2151	2153							
@@E024	001	0015	2153	2155							
@@E025	001	0016	2155	2157							
@@E026	001	0017	2157	2159							
@@E027	001	0018	2159	2161							
@@E028	001	0019	2161	2163							
@@E029	001	001A	2163	2165							
@@E030	001	001B	2165	2167							
@@E031	001	001C	2167	2169							
@@E032	001	001D	2169	2171							
@@E035	001	001E	2171	2173							
@@E036	001	001F	2173	2175							
@@E037	001	0020	2175	2177							
@@E038	001	0021	2177	2179							
@@E039	001	0022	2179	2181							
@@E040	001	0023	2181	2183							
@@E041	001	0024	2183	2185							
@@E042	001	0025	2185	2187							
@@E043	001	0026	2187	2189							
@@E044	001	0027	2189	2191							
@@E045	001	0028	2191	2193							
@@E046	001	0029	2193	2195							
@@E060	001	002A	2195	2197							
@@E080	001	002B	2197								
@@E100	001	0000	1583	1585							
@@E101	001	0001	1585	1587							
@@E102	001	0002	1587	1589							

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 283

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E103	001	0003	1589	1591
@@E110	001	0004	1591	1593
@@E112	001	0005	1593	1595
@@E113	001	0006	1595	1597
@@E114	001	0007	1597	1599
@@E115	001	0008	1599	1601
@@E116	001	0009	1601	1603
@@E117	001	000A	1603	1605
@@E120	001	000B	1605	1607
@@E122	001	000C	1607	1609
@@E123	001	000D	1609	1611
@@E124	001	000E	1611	1613
@@E129	001	000F	1613	1615
@@E130	001	0010	1615	1617
@@E131	001	0011	1617	1619
@@E133	001	0012	1619	1621
@@E134	001	0013	1621	1623
@@E135	001	0014	1623	1625
@@E136	001	0015	1625	1627
@@E137	001	0016	1627	1629
@@E138	001	0017	1629	1631
@@E139	001	0018	1631	1633
@@E142	001	0019	1633	1635
@@E143	001	001A	1635	1637
@@E150	001	001B	1637	1639
@@E151	001	001C	1639	1641
@@E160	001	001D	1641	1643
@@E162	001	001E	1643	1645
@@E163	001	001F	1645	1647
@@E164	001	0020	1647	1649
@@E200	001	0021	1649	1651
@@E205	001	0022	1651	1653
@@E210	001	0023	1653	1655
@@E211	001	0024	1655	1657
@@E212	001	0025	1657	1659
@@E213	001	0026	1659	1661
@@E215	001	0027	1661	1663
@@E216	001	0028	1663	1665
@@E217	001	0029	1665	1667
@@E220	001	002A	1667	1669
@@E221	001	002B	1669	1671
@@E222	001	002C	1671	1673
@@E223	001	002D	1673	1675
@@E225	001	002E	1675	1677
@@E226	001	002F	1677	1679
@@E227	001	0030	1679	1681
@@E228	001	0031	1681	1683
@@E229	001	0032	1683	1685
@@E230	001	0033	1685	1687
@@E232	001	0034	1687	1689
@@E234	001	0035	1689	1691
@@E237	001	0036	1691	1693
@@E240	001	0037	1693	1695
@@E241	001	0038	1695	1697 2709
@@E242	001	0039	1697	1699
@@E248	001	003A	1699	1701



## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 284

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E249	001	003B	1701	1703
@@E250	001	003C	1703	1705
@@E251	001	003D	1705	1707
@@E252	001	003E	1707	1709
@@E253	001	003F	1709	1711
@@E254	001	0040	1711	1713
@@E255	001	0041	1713	1715
@@E256	001	0042	1715	1717
@@E300	001	0043	1717	1719
@@E301	001	0044	1719	1721
@@E302	001	0045	1721	1723
@@E303	001	0046	1723	1725
@@E304	001	0047	1725	1727
@@E305	001	0048	1727	1729
@@E308	001	0049	1729	1731
@@E310	001	004A	1731	1733
@@E315	001	004B	1733	1735
@@E316	001	004C	1735	1737
@@E320	001	004D	1737	1739
@@E325	001	004E	1739	1741
@@E330	001	004F	1741	1743
@@E335	001	0050	1743	1745
@@E338	001	0051	1745	1747
@@E340	001	0052	1747	1749
@@E350	001	0053	1749	1751
@@E351	001	0054	1751	1753
@@E352	001	0055	1753	1755
@@E360	001	0056	1755	1757
@@E361	001	0057	1757	1759
@@E362	001	0058	1759	1761
@@E371	001	0059	1761	1763
@@E380	001	005A	1763	1765
@@E390	001	005B	1765	1767
@@E400	001	005C	1767	1769
@@E410	001	005D	1769	1771
@@E415	001	005E	1771	1773
@@E417	001	005F	1773	1775
@@E420	001	0060	1775	1777
@@E430	001	0061	1777	1779
@@E432	001	0062	1779	1781
@@E433	001	0063	1781	1783
@@E450	001	0064	1783	1785
@@E451	001	0065	1785	1787
@@E460	001	0066	1787	1789
@@E461	001	0067	1789	1791
@@E464	001	0068	1791	1793
@@E465	001	0069	1793	1795
@@E466	001	006A	1795	1797
@@E467	001	006B	1797	1799
@@E469	001	006C	1799	1801
@@E470	001	006D	1801	1803
@@E471	001	006E	1803	1805
@@E473	001	006F	1805	1807
@@E474	001	0070	1807	1809
@@E475	001	0071	1809	1811
@@E476	001	0072	1811	1813

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 285

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E477	001	0073	1813	1815
@@E478	001	0074	1815	1817
@@E479	001	0075	1817	1819
@@E480	001	0076	1819	1821
@@E481	001	0077	1821	1823
@@E482	001	0078	1823	1825
@@E483	001	0079	1825	1827
@@E484	001	007A	1827	1829
@@E485	001	007B	1829	1831
@@E486	001	007C	1831	1833
@@E487	001	007D	1833	1835
@@E488	001	007E	1835	1837
@@E489	001	007F	1837	1839
@@E490	001	0080	1839	1841
@@E491	001	0081	1841	1843
@@E492	001	0082	1843	1845
@@E493	001	0083	1845	1847
@@E494	001	0084	1847	1849
@@E495	001	0085	1849	1851
@@E496	001	0086	1851	1853
@@E497	001	0087	1853	1855
@@E498	001	0088	1855	1857
@@E500	001	0089	1857	1859
@@E501	001	008A	1859	1861
@@E530	001	008B	1861	1863
@@E531	001	008C	1863	1865
@@E535	001	008D	1865	1867
@@E540	001	008E	1867	1869
@@E541	001	008F	1869	1871
@@E542	001	0090	1871	1873
@@E543	001	0091	1873	1875
@@E544	001	0092	1875	1877
@@E545	001	0093	1877	1879
@@E546	001	0094	1879	1881
@@E547	001	0095	1881	1883
@@E548	001	FFFF	2087	
@@E549	001	0096	1883	1885
@@E550	001	0097	1885	1887
@@E551	001	0098	1887	1889
@@E552	001	0099	1889	1891
@@E553	001	009A	1891	1893
@@E554	001	009B	1893	1895
@@E555	001	009C	1895	1897
@@E556	001	009D	1897	1899
@@E558	001	009E	1899	1901
@@E570	001	009F	1901	1903
@@E571	001	00A0	1903	1905
@@E572	001	00A1	1905	1907
@@E573	001	00A2	1907	1909
@@E574	001	00A3	1909	1911
@@E575	001	FFFF	2089	
@@E578	001	00A4	1911	1913
@@E579	001	FFFF	2091	
@@E580	001	FFFF	2093	
@@E585	001	00A5	1913	1915
@@E595	001	FFFF	2095	

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 286

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E597	001	FFFF	2097	
@@E598	001	FFFF	2099	
@@E600	001	00A6	1915	1917
@@E601	001	00A7	1917	1919
@@E602	001	00A8	1919	1921
@@E603	001	00A9	1921	1923
@@E604	001	00AA	1923	1925
@@E606	001	00AB	1925	1927
@@E607	001	00AC	1927	1929
@@E608	001	00AD	1929	1931
@@E609	001	00AE	1931	1933
@@E610	001	00AF	1933	1935
@@E611	001	00B0	1935	1937
@@E612	001	00B1	1937	1939
@@E613	001	00B2	1939	1941
@@E614	001	00B3	1941	1943
@@E700	001	00B4	1943	1945
@@E701	001	00B5	1945	1947 7292
@@E710	001	00B6	1947	1949 7381 7385 7403
@@E712	001	00B7	1949	1951 7579 8150
@@E713	001	00B8	1951	1953
@@E714	001	00B9	1953	1955 8406
@@E715	001	00BA	1955	1957 7743
@@E716	001	00BB	1957	1959
@@E717	001	00BC	1959	1961
@@E718	001	00BD	1961	1963 8258 0554
@@E720	001	00BE	1963	1965 1692
@@E721	001	00BF	1965	1967 1752
@@E723	001	00C0	1967	1969
@@E724	001	00C1	1969	1971
@@E725	001	00C2	1971	1973
@@E726	001	00C3	1973	1975
@@E727	001	00C4	1975	1977
@@E728	001	00C5	1977	1979 7258
@@E729	001	00C6	1979	1981
@@E730	001	00C7	1981	1983
@@E732	001	00C8	1983	1985 7268
@@E752	001	00C9	1985	1987
@@E753	001	00CA	1987	1989
@@E754	001	00CB	1989	1991
@@E755	001	00CC	1991	1993
@@E756	001	00CD	1993	1995
@@E757	001	00CE	1995	1997
@@E758	001	00CF	1997	1999
@@E759	001	00D0	1999	2001
@@E760	001	00D1	2001	2003
@@E761	001	00D2	2003	2005
@@E762	001	00D3	2005	2007
@@E763	001	00D4	2007	2009
@@E764	001	00D5	2009	2011
@@E765	001	00D6	2011	2013
@@E766	001	00D7	2013	2015
@@E767	001	00D8	2015	2017
@@E768	001	00D9	2017	2019
@@E769	001	00DA	2019	2021
@@E770	001	00DB	2021	2023

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 287

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E771	001	00DC	2023	2025
@@E772	001	00DD	2025	2027
@@E773	001	00DE	2027	2029
@@E774	001	00DF	2029	2031 5160
@@E775	001	00E0	2031	2033 5364
@@E776	001	00E1	2033	2035 4806
@@E777	001	00E2	2035	2037 4128
@@E778	001	00E3	2037	2039 4108
@@E779	001	00E4	2039	2041 4119
@@E780	001	00E5	2041	2043 6193
@@E781	001	00E6	2043	2045
@@E782	001	00E7	2045	2047
@@E783	001	00E8	2047	2049
@@E784	001	00E9	2049	2051
@@E785	001	00EA	2051	2053
@@E786	001	00EB	2053	2055
@@E790	001	00EC	2055	2057 5124
@@E791	001	00ED	2057	2059 5235
@@E792	001	00EE	2059	2061
@@E793	001	00EF	2061	2063 6205
@@E794	001	00F0	2063	2065 6196
@@E795	001	00F1	2065	2067 5613
@@E796	001	00F2	2067	2069 5599
@@E797	001	00F3	2069	2071
@@E798	001	00F4	2071	2073
@@E800	001	FFFF	2101	
@@E801	001	FFFF	2103	
@@E802	001	FFFF	2105	
@@E803	001	FFFF	2107	
@@E804	001	FFFF	2109	
@@E900	001	00F5	2073	2075 2705
@@E901	001	00F6	2075	2077 2707
@@E902	001	00F7	2077	2079 2706
@@E903	001	00F8	2079	2081 2708
@@E905	001	00F9	2081	2083
@@E906	001	00FA	2083	2085
@@E910	001	00FB	2085	2704
@@M250	001	2D00	0361	0170
@@M251	001	2D04	0365	0179
@@M256	001	2D08	0369	0191
@@M257	001	2D0C	0373	
@@M258	001	2D10	0377	
@@M259	001	2D14	0381	
@@M260	001	2D18	0385	
@@T250	001	2D1C	0389	0363
@@T251	001	2D24	0391	0367
@@T256	001	2D2D	0393	0371
@@T257	001	2D52	0395	0375
@@T258	001	2D75	0397	0379
@@T259	001	2D8C	0399	0383
@@T260	001	2DAC	0401	0387
@ALTFL	001	0001	0963	
@ARR	001	0008	0017	6800 7765 9142 9149 9163 9182 0233 0586 0629 0804 1004 1213 1446 1466 2645 2662 2919 2937
@ASIGN	001	007C	0072	
@ASTER	001	005C	0070	

CROSS REFERENCE																		
SYMBOL	LEN	VALUE	DEFN	REFERENCES										VER 15, MOD 00			31/05/21	PAGE 288
@BCRDL	001	0050	0089	7948														
@BE	001	0081	0044	0489														
@BF	001	0090	0053															
@BH	001	0084	0042															
@BKSPC	001	0010	1060															
@BL	001	0082	0043															
@BLANK	001	0040	0066	7371	8188													
@BM	001	0082	0055															
@BNE	001	0001	0047	8174	8175	0488												
@BNH	001	0004	0045															
@BNL	001	0002	0046															
@BNM	001	0002	0058															
@BNOL	001	0020	0051															
@BNOZ	001	0008	0050															
@BNP	001	0004	0057															
@BNZ	001	0001	0059															
@BOL	001	00A0	0049															
@BOZ	001	0088	0048															
@BP	001	0084	0054															
@BR	001	0001	0014	4099	4107	4107	4108	4109	4118	4118	4119	4120	4121	4127	4128			
				4134	4136	4138	4175	4176	4177	4178	4180	4181	4186	4187	4212			
				4213	4244	4253	4254	4258	4267	4267	4268	4273	4273	4279	4280			
				4297	4301	4301	4305	4306	4310	4321	4327	4327	4332	4332	4333			
				4333	4334	4479	4487	4488	4491	4504	4511	4620	4627	4629	4643			
				4645	4645	4652	4653	4656	4667	4668	4790	4805	4807	4821	4827			
				4828	4832	4833	4834	4835	4836	4836	4837	4846	4847	4850	4854			
				4855	4856	4857	4857	4858	4884	4884	4885	4895	4905	4906	4907			
				4909	4909	4916	4916	4917	4926	4928	4936	4936	4939	4981	4984			
				4986	4987	5099	5113	5118	5123	5125	5130	5132	5133	5142	5144			
				5145	5145	5146	5147	5149	5151	5152	5153	5159	5161	5168	5170			
				5175	5175	5176	5180	5181	5190	5192	5193	5201	5202	5203	5208			
				5209	5210	5220	5226	5234	5236	5240	5357	5363	5365	5367	5375			
				5381	5385	5398	5407	5408	5408	5409	5410	5411	5412	5413	5422			
				5423	5424	5424	5425	5426	5431	5432	5439	5440	5440	5441	5441			
				5443	5584	5591	5596	5599	5600	5611	5612	5613	5615	5616	5623			
				5625	5629	5635	5647	5651	5652	5653	5654	5655	5656	5657	5658			
				5669	5685	5686	5687	5688	5689	5690	5693	5694	5699	5700	5701			
				5730	5740	5741	5744	5745	5745	5746	5746	5751	5751	5752	5753			
				5759	5761	5762	5762	5763	5764	5892	5910	5919	5920	5920	5921			
				6035	6042	6049	6051	6058	6058	6059	6186	6192	6193	6194	6196			
				6197	6198	6199	6205	6207	6209	6210	6214	6223	6337	6343	6344			
				6351	6352	6353	6357	6358	6363	6388	6389	6390	6396	6422	6427			
				6427	6428	6429	6436	6436	6437	6439	6446	6447	6448	6449	6450			

## CROSS REFERENCE

SYMBOL   LEN   VALUE   DEFN   REFERENCES   VER 15, MOD 00   31/05/21   PAGE 289

	7887	7888	7894	7895	7907	7908	7914	7915	7916	7921	7921	7922
	7925	7926	7926	7927	7927	7928	7940	7944	7944	7945	7945	7948
	7949	7965	7979	7980	7984	7984	7989	7996	7996	8000	8000	8001
	8001	8009	8010	8011	8012	8137	8139	8163	8165	8170	8175	8183
	8184	8197	8210	8217	8218	8224	8226	8231	8235	8236	8240	8242
	8254	8259	8268	8289	8301	8311	8312	8315	8316	8316	8317	8318
	8319	8320	8328	8331	8333	8333	8334	8334	8336	8336	8338	8339
	8339	8340	8340	8342	8344	8346	8346	8357	8358	8359	8359	8371
	8372	8373	8376	8376	8377	8410	8437	8461	8461	8462	8469	8469
	8470	8472	8472	8473	8484*	8499	8506	8507	8508	8515	8516	8517
	8521	8522	8523	8529	8530	8530	8535	8537	8538	8539	8541	8542
	8543	8543	8544	8552	8557	8563	8564	8569	8572	8622	8628	8629
	8635	8645	8663	8688	8696	8696	8698	8701	8702	8706	8708	8710
	8711	8729	8733	8734	8744	8912	8916	8917	8917	8918	8922	8923
	8924	8925	8926	8927	8928	8929	8930	8930	8931	8932	8933	8934
	8935	8938	8940	8944	8945	8949	8951	8953	9039	9040	9041	9041
	9042	9103	9105	9105	9111	9114	9115	9118	9120	9121	9122	9124
	9124	9125	9127	9128	9141	9142	9143	9143	9144	9145	9149	9150
	9150	9151	9151	9152	9153	9163	9164	9165	9170	9176	9182	9183
	9183	9197	9198	9199	9201	9203	9205	9206	9208	9210	9211	9217
	9221	9222	9224	9224	9225	9225	9227	9237	9237	9240	9241	9241
	9242	9251	9254	9256	9256	9257	9263	9264	9264	9265	9268	9269
	9282	9283	9284	9293	9299	9314	9316	9318	9321	9323	9328	9330
	9331	9333	9338	9339	9340	9342	9344	9344	9346	9347	9347	9348
	9348	9350	9351	9351	9354	9354	9355	9357	9359	9361	9363	9363
	9364	9364	9365	9365	9367	9383	9383	9384	9386	9387	9388	9389
	9393	9401	9403	9404	9408	9409	9462	9478	9488	9488	9490	9493
	9493	9495	9495	9497	9498	9500	9501	9502	9502	9504	9509	9512
	9517	9522	9536	9964	9978	0008	0011	0012*	0031	0055	0097	0111
	0120	0122	0126	0128	0129	0143	0144	0144	0145	0145	0146	0147
	0149	0149	0150	0150	0152	0152	0153	0154	0154	0155	0159	0163
	0166	0171	0173	0174	0180	0182	0183	0184	0185	0190	0193	0194
	0195	0201	0224	0228	0233	0237	0237	0238	0242	0247	0254	0342
	0441	0457	0458	0459	0460	0473	0474	0485	0486	0489	0490	0492
	0493	0494	0502	0507	0508	0509	0524	0528	0529	0531	0545	0561
	0586	0595	0595	0602	0603	0603	0605	0606	0608	0609	0611	0611
	0612	0614	0629	0633	0692	0698	0699	0717	0729	0754	0757	0760
	0771	0771	0773	0776	0776	0777	0778	0779	0780	0784	0784	0786
	0786	0788	0788	0804	0808	0861	0867	0867	0868	0869	0870	0880
	0881	0886	0887	0889	0889	0890	0891	0903	0903	0904	0905	0910
	0912	0913	0932	0933	0938	0939	0948	0949	0951	0951	0952	0953
	0957	0958	0959	0959	0960	0960	0961	0961	0962	0962	0963	0963
	0964	0969	0969	0971	0971	0981	0981	0983	0983	1004	1008	1012
	1068	1080	1082	1092	1093	1098	1100	1112*	1127	1128	1128	1129
	1143	1144	1146	1146	1147	1148	1148	1149	1154	1158	1160	1174
	1175	1176	1180	1182	1183	1195	1197	1201*	1202	1213	1217	1301
	1307	1308	1308	1309	1310	1321	1322	1326	1333	1333	1334	1335
	1339	1348	1348	1350	1352	1353	1353	1355	1356	1362	1363	1367
	1369	1382	1383	1388	1389	1389	1390	1391	1392	1397	1397	1398
	1398	1399	1404	1404	1406	1406	1411	1413	1414	1418	1420	1446
	1448	1453	1466	1470	1681	1699	1706	1707	1708	1709	1714	1734
	1758	1771	1773	2029	2040	2043	2048	2058	2061	2063	2070	2070
	2082	2083	2083	2084	2089	2089	2090	2091	2096	2096	2106	2112
	2128	2130	2131	2131	2134	2136	2136	2143	2146	2160	2161	2161
	2169	2170	2179	2272	2279	2282	2287	2289	2297	2298	2305	2306
	2306	2307	2307	2312	2313	2313	2315	2319	2335	2341	2342	2342

## CROSS REFERENCE

SYMBOL   LEN   VALUE   DEFN   REFERENCES   VER 15, MOD 00   31/05/21   PAGE 290

2344	2350	2354	2354	2356	2357	2357	2363	2364	2365	2369	2369		
2370	2371	2373	2373	2374	2374	2375	2375	2376	2380	2382	2382		
2383	2387	2388	2388	2389	2445	2456	2460	2461	2462	2467	2467		
2469	2473	2484	2486	2488	2497	2510	2512	2512	2514	2514	2515		
2516	2518	2518	2526	2527	2533	2534	2534	2545	2550	2559	2568		
2572	2572	2573	2574	2574	2586	2595	2596	2605	2606	2616	2620		
2621	2645	2649	2650	2655	2662	2730	2736	2740	2744	2745	2746		
2751	2751	2753	2757	2768	2780	2793	2795	2795	2797	2797	2798		
2799	2801	2801	2809	2810	2816	2817	2817	2828	2833	2842	2851		
2855	2855	2856	2857	2857	2869	2878	2879	2888	2889	2899	2903		
2905	2919	2923	2924	2929	2937	2939	3132	3142	3148	3153	3154		
3154	3161	3162	3162	3163	3176	3196	3198	3198	3199	3200	3210		
3211	3211	3218	3218	3219	3219	3220	3220	3221	3222	3222	3232		
3232	3236	3236	3237	3237	3241	3243	3247	3249	3259	3259	3260		
3260	3264	3265	3272	3272	3273	3429	3433	3440	3442	3444	3453		
3468	3469	3487	3490	3497	3498	3499	3530	3531	3532	3533	3535		
3542	3572	3586	3621	3627	3631	3636	3637	3648	3649	3650	3650		
3651	3651	3652	3652	3653	3653	3654	3654	3659	3665	3667	3677		
3686	3688	3689	3692	3694	3695	3696	3697	3699	3733	3735	3739		
3741	3741	3743	3744	3744	3746	3748	3756	3757	3766	3767	3768		
3768	3770	3774	3775	3778	3786	3788	3788	3791	3791	3792	3793		
3795	3800	3801	3802										

@BT   001   0010   0052  
@BZ   001   0081   0056  
@BZ37B   001   00F2   1073  
@B1   001   0001   0064

4238	4354	4679	4954	5257	5472	5778	6049	6493	7400	7641	7647				
7718	7724	7735	7749	7750	7751	7757	7759	7922	7995	8204	8212				
8230	8392	8502	0146	0153	0631	0750	0755	0806	1006	1133	1136				
1215	1468	2063	2150	2370	3667										
@CADDR	001	0002	0143	2658	2685	3502	3503	3504	7039	7040	7041	7042	7043	7044	7045
				7046	7047	7248	7263	7302	7311	7336	7363	7585	7588	7592	7608
				7655	7671	7747	7768	7794	7812	7833	7848	8026	8276	8328	8424
				8426	8427	8428	8430	8483	8564	8930	9074	9105	9150	9151	9152
				9170	9183	9237	9241	9256	9261	9264	9316	9318	9321	9348	9365
				9422	9495	9498	9540	9980	0008	0011	0014	0065	0237	0250	0254
				0276	0363	0367	0371	0375	0379	0383	0387	0492	0502	0603	1790
				1791	1792	2179	2203	2451	2462	2488	2736	2746	2941	3154	3259
				3267	3278	3290	3299	3475	3513	3741					

@CARDL   001   0060   0088  
@CC37B   001   0000   1069  
@CD37B   001   00F0   1087  
@CHARA   001   00C1   0073  
@CHARF   001   00C6   0074  
@CHARR   001   00D9   0075  
@CHARZ   001   00E9   0076  
@CKY01   001   0001   1021  
@CKY02   001   0002   1022  
@CKY03   001   0003   1023  
@CKY04   001   0004   1024  
@CKY05   001   0005   1025  
@CKY06   001   0006   1026  
@CKY07   001   0007   1027  
@CKY08   001   0008   1028  
@CKY09   001   0009   1029  
@CKY10   001   000A   1030  
@CKY11   001   000B   1031

0645   7951   7966\*   7967   7967   7967\*   8173   8182\*



## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 291

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@CKY12	001	000C	1032	
@CKY13	001	000D	1033	
@CKY14	001	000E	1034	
@CKY15	001	000F	1035	
@CKY16	001	0010	1036	
@CLOFF	001	0010	0095	
@CLON	001	0011	0094	
@CMLON	001	0001	1039	
@CMOFF	001	0000	1038	
@COMMA	001	006B	0067	8191 8221
@CPLUS	001	004E	0080	
@CP37B	001	0004	1100	
@CRERR	001	0090	1055	
@CRPRY	001	0004	1059	
@CRTDS	001	0092	1052	
@CRTQ	001	0090	1054	
@CURSR	001	0040	1056	
@DADDR	001	0002	0141	3218 3219 3293 3301
@DBFR1	001	0004	0130	3260*
@DBFR2	001	0005	0131	3259*
@DBUSY	001	0002	0957	
@DCALK	001	0001	0082	
@DCBCY	001	0009	0116	3331
@DCBT1	001	0050	0118	3334
@DCFLN	001	0004	0941	
@DCNT	001	0003	0129	
@DCRID	001	0001	0955	
@DCST1	001	0040	0117	3332
@DCTRL	001	0000	0126	3142* 3148* 3176
@DCTRW	001	0000	0954	
@DCWID	001	0001	0951	
@DCYL	001	0001	0127	3210* 3232*
@DCYMV	001	0001	0942	
@DD2	001	0003	0031	5746 5746* 5751* 6427 6427* 6436* 6763* 1707* 3563
@DEFLG	001	0002	0964	
@DERCE	001	0020	0994	
@DERD2	001	0008	0986	
@DEREQ	001	0010	0985	
@DERIN	001	0040	0983	
@DERMA	001	0020	0984	
@DERNR	001	0004	0987	
@DERR	001	0000	0958	
@DERSC	001	0001	0989	
@DERTC	001	0002	0988	
@DFCR	001	0006	0944	
@DFDR	001	0004	0945	
@DGET	001	0001	0135	7865 8471 8594 3148 3176
@DHARD	001	0000	0972	
@DLNCT	001	000F	1058	
@DLNLG	001	0040	1057	2740
@DOLAR	001	005B	0069	
@DOP2	001	0004	0029	8005 9445 0649 3792*
@DPLNG	001	0006	0133	
@DPOS	001	0000	0134	
@DPUT	001	0002	0136	7813 7855 8582 3142
@DREAD	001	0001	0948	

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 292

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@DSAD	001	0002	0128	3211* 3220* 3222* 3236 3236* 3237 3237* 3243* 3249*
@DSBCY	001	0004	0107	3269
@DSBSY	001	0092	1053	
@DSCS1	001	0000	0108	3270
@DSEEK	001	0000	0947	
@DSIVF	001	0003	0139	
@DSPIN	001	0002	0132	
@DTRSZ	001	0018	0086	3292
@DUNSF	001	0080	0990	
@DVBCY	001	0007	0109	3328
@DVERY	001	0003	0953	
@DVRFY	001	0031	0137	
@DVST1	001	0002	0959	
@DVST2	001	0003	0960	
@DWAIT	001	00FF	0138	
@DWBCY	001	0005	0104	3325
@DWRIT	001	0002	0949	
@DWSIZ	001	00C0	0106	
@DWTB1	001	0003	0105	3326
@DZERO	001	00F0	0065	4133 4172 4251 4489 4636 4866 4927 5112 5117 5129 5366 5380 5387 5407 5663 5900 6191 6484
@D1	001	0002	0027	6765* 8003 8183* 8217 8218* 8224* 8235* 9398 9509* 1129* 1143 1148* 1227 1309* 1350 1353* 1481 1709* 2084* 2089* 2090 2096 2130* 2131* 2136 2305* 2306* 2307* 2312* 2313* 2319 2341* 2342* 2350 2387* 2388 2388* 2473* 2757* 3153* 3161* 3162* 3196* 3198* 3199 3211 3260 3272* 3440* 3490* 3562 3631* 3794* 3795*
@EOF	001	001C	0078	7704 8323 8665
@EOFTC	001	0075	0163	
@EOS	001	001E	0077	3341 8182 8194 8206 9255
@ER37B	001	00F0	1074	
@FDDBC	001	0000	0196	
@FDE1	001	000C	0201	
@FDFNA	001	000B	0199	
@FDHLN	001	0002	0209	
@FDLNC	001	0002	0194	
@FDNSC	001	0003	0211	
@FDSD	001	0000	0207	
@FLACE	001	0009	0198	
@FLDBC	001	0001	0197	
@FLDIN	001	0012	1046	
@FLENT	001	0004	0202	
@FLFNA	001	0002	0200	
@FLHLN	001	0002	0210	
@FLLNC	001	0002	0195	
@FLNSC	001	0001	0212	
@FLSD	001	0001	0208	
@HCEPK	001	003C	0829	
@HCOPS	001	001C	0836	
@HCOPY	001	081C	0831	
@HCRHE	001	7858	0852	
@HDNRY	001	1008	0817	
@HDRHE	001	7854	0850	
@HDRLN	001	0007	0093	0673
@HDRV1	001	7840	0842	
@HDRV2	001	7844	0844	
@HDTRD	001	1040	0813	

CROSS REFERENCE																		
SYMBOL	LEN	VALUE	DEFN	REFERENCES												VER 15, MOD 00	31/05/21	PAGE 293
@HDTRJ	001	1010	0815															
@HERPG	001	087C	0819															
@HFEHT	001	0804	0834															
@HIPLE	001	006C	0826															
@HKBER	001	2040	0809	9297														
@HKBHE	001	7848	0846															
@HLOGE	001	1844	0821															
@HPRER	001	0070	0811	9477														
@HPRHE	001	784C	0848															
@HSTAD	001	0009	0970															
@HSTEN	001	0007	0969															
@HSTPE	001	0006	0968	9478*	9528*													
@HSTQR	001	0001	0966															
@HSTSN	001	0005	0967															
@HSTVI	001	000F	0971															
@HUNSF	001	1850	0824															
@IAR	001	0010	0018															
@ID37B	001	0040	1110															
@INDEX	001	0001	0157	0158	8739	9361	9465	9468	9507	3460	3462	3497						
@INST3	001	0003	0033	7952	9372	9400	3507	3520										
@INST4	001	0004	0034	6427	6436	6469	6477	6489	6492	6815	2203							
@INST5	001	0005	0035															
@INST6	001	0006	0036															
@IP37B	001	00C0	1109															
@I1IAR	001	00C0	0021	8945*	9103*													
@KCMDK	001	0020	1020															
@KELOK	001	001B	1019															
@KENAB	001	001E	1017	8630	8707	8946	9054	9172	9175	9994	0058	0259	3745					
@KEXIT	001	001F	1018															
@KEYBD	001	0010	1037	8946	9036	9040	9053	9104	9108	9110	9185	9223	9227					
@KFUNK	001	0010	1040	9073	9118													
@KHARD	001	0011	1045															
@KLEAR	001	000D	1041															
@LINSZ	001	00F4	0085	0647														
@LO37B	001	00F0	1078															
@MAPEN	001	0005	0090															
@MINCR	001	2000	0084															
@MINUS	001	0060	0081															
@NOP	001	0080	0041	3844	4120	5203	6357	6814	6878	7722	8163	8197	8312	8386	8396			
				8528	8541	8562	8569	8628	8629	8674	8706	9123	9216	9234	9294			
				9371	9373	9397	9520	0005	0010	0455	0460	0473	0490	0868	0887			
				0910	0932	1012	1326	1367	1448	3452	3511	3519	3530	3531	3756			
@NORFL	001	0000	0965															
@NTRDY	001	00A0	1102															
@NUMBR	001	007B	0071															
@OPD2	001	0004	0030	6428*	6469*	6470*	6477	6477*	9124*	2170*								
@OP1	001	0003	0028	4273*	5919*	5920*	5921*	6800*	7392*	7425*	7660	7731	7753	7765*	7796*			
				7823*	7824*	7886*	8009	8226*	8349	8356	8485	8487	8663*	8700	8733*			
				8917*	8918*	9142*	9149*	9151*	9152*	9163*	9182*	9318*	0011*	0166*	0190			
				0233*	0237*	0238*	0242	0254	0279	0586*	0629*	0804*	1004*	1080*	1092*			
				1158*	1176*	1213*	1418*	1446*										

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 294

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@PCNT	001	0003	0939	
@PCTRL	001	0000	0150	9233* 9235* 9328 9331 9346* 9355 9361* 9500* 9504* 2692 2971 3453 3487 3800*
@PCYL	001	0001	0937	
@PC37B	001	00F2	1094	
@PDAR	001	00E4	1003	9338* 3626*
@PDATA	001	0003	0152	9065 9150* 9183 9260* 9261* 9338 9348* 9495* 0122* 0183* 0228* 2694 2973 3741 3791* 3792
@PD37B	001	0080	1108	
@PERR	001	00E0	1010	8744 9389 9391 9393 3572 3574
@PFLAG	001	0000	0936	
@PFORM	001	00E1	1008	9387
@PGCSZ	001	0020	0083	0084
@PLITE	001	00E2	1009	9388* 9408*
@PLNGH	001	0004	1000	3739 3739 3739*
@PMGCK	001	0020	1011	9479
@PN37B	001	00F0	1093	
@PPLNG	001	0004	0149	0191 0192
@PRCNT	001	0001	0151	9330* 9339* 9340* 9342* 9344 9344* 9347 9350 9351* 9354* 2693 2972 3468 3469 3735 3744 3744* 3801*
@PRETR	001	00C0	0155	0302 0369 0373 0377 0381 0385 2550 2833 3487 3774
@PRINT	001	0040	0153	0155 9328 9485 9505 9529 0309 0316 0361 0365 2460 2744 3435 3453 3534 3778 3800
@PRITY	001	0080	1044	9115
@PSAD	001	0002	0938	
@PSIOQ	001	00E0	1006	9368 9541
@PSIOR	001	0000	1005	9368 9542
@PSNSQ	001	00E2	1012	9470
@PSR	001	0004	0016	
@PWAIT	001	00FF	0159	7418
@P1IAR	001	0020	0019	9039* 9111*
@P2IAR	001	0040	0020	
@Q	001	0001	0025	4108* 4119* 4120* 4127* 4128* 5203* 5209* 5599* 5613* 6049* 6058 6058* 6193* 6196* 6205* 6352* 6357* 6813 6876* 6878* 7418* 7445* 7754 7922* 7926 7926* 7950 8004 8163* 8175* 8184* 8197* 8312* 8315* 8340* 8537* 8541* 8569* 8572* 8628* 8629* 8635* 8706* 8710* 8711* 9117* 9123* 9216* 9236* 9294* 9370 9384* 9396 9520* 0146* 0150 0150* 0153* 0457* 0458* 0459* 0460* 0473* 0474* 0485* 0486* 0489* 0490* 0561* 0605* 0608* 0614* 0698* 0699* 0868* 0869* 0870* 0886* 0887* 0910* 0932* 0938* 1012* 1326* 1339* 1367 1448* 1453* 2058* 2061* 2082* 2083* 2160* 2161* 2246 2297* 2298* 2370* 2374 2374* 2682 2683 2703 2961 2962 3433* 3441* 3452* 3498* 3505 3508* 3518 3530* 3531* 3621* 3756*
@RD37B	001	00F1	1088	
@REGL	001	0002	0013	5256 7330 9041
@RETRN	001	0080	0154	0155 9346 9355 9431 9500 9504 0323 2698 2977
@RLDWN	001	004F	0160	
@RTCNT	001	0003	1002	9357* 9359* 9363*
@RTRNC	001	0080	0162	0324 2699 2978
@RT37B	001	0005	1101	
@SBLN	001	0005	0171	
@SBLNL	001	0002	0185	
@SCTS	001	0100	0101	
@SDFLN	001	0007	0091	
@SDF0	001	0000	0167	8512
@SDF1	001	0001	0168	8530* 8538 8543* 8544
@SDF2	001	0002	0169	8523

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 31/05/21 PAGE 295

@SDF3	001	0003	0170												
@SECCY	001	0030	0087												
@SIST	001	0001	0182												
@SKCTL	001	0000	0952												
@SLASH	001	0061	0068												
@SLAST	001	0002	0184	8523											
@SMIDL	001	0003	0183												
@SNSB0	001	0000	0976												
@SNSB1	001	0001	0977												
@SNSB2	001	0002	0978												
@SNSB3	001	0003	0979												
@SNULL	001	0080	0174	8512											
@SN37B	001	00F2	1082												
@SONLY	001	0000	0181												
@SPINA	001	00A0	0961												
@SPINB	001	00B0	0962												
@STEXT	001	0007	0173	8610											
@STYPE	001	0006	0172	8535											
@SYCNT	001	0002	1001	9490*	9501	9502*	3743*								
@SYLVL	001	0005	2740												
@TBCNT	001	0000	0161												
@TBLEF	001	0010	0156	0158	9331										
@TBLIX	001	0011	0158	3533											
@TJ37B	001	0040	1099												
@TYPAM	001	0002	1043	9221	9251										
@TYPO	001	001C	1042												
@UCB	001	0087	0040	3821	4127	5209	6352	6876	7722	8184	8242	8315	8396	8537	8572
				8635	8649	8695	8710	8711	9117	9236	9289	9384	0457	0458	0459
				0485	0553	0561	0869	0870	0886	0938	1339	1453	3433	3441	3498
				3506	3508	3621									
@UPARW	001	005A	0079	2723	8976										
@VADDR	001	0002	0142	2659	2685	3062	3498	3510	3511	3512	3512	3526	3529	3531	3555
				3556	3557	3595	3598	3601	3604	3607	3610	3613	3622	3625	3628
				3631	3634	4154	4162	4207	4502	4513	4798	4815	4878	5215	5232
				5640	5665	5697	6371	6383	6395	6837	6840	6846	6850	6883	6905
				6922	6944	6966	6988	7005	7022	7242	7253	7274	7282	7299	7300
				7301	7314	7315	7320	7321	7342	7360	7361	7397	7419	7442	7443
				7452	7453	7460	7568	7612	7621	7663	7767	7875	7961	7973	7986
				8139	8167	8180	8247	8252	8254	8264	8268	8274	8307	8322	8369
				8375	8410	8507	8510	8555	8567	8644	8668	8698	8708	8736	8949
				8951	9324	9401	9514	9978	9992	0000	0009	0020	0029	0047	0055
				0065	0066	0163	0201	0261	0278	0514	0732	1115	1188	1257	1261
				1265	1269	1273	1277	1699	1702	1734	2104	2185	2486	2631	2664
				3436	3473	3491	3509	3521	3545	3589	3750	3777	3782		
@VENTA	001	0056	0114	3329	3584	7251	7287	1687	1747						
@VMDDV	001	00FE	0115												
@VMFD1	001	0000	0110												
@VMFD2	001	0001	0111												
@VMRS3	001	0002	0113												
@VMTRL	001	0001	0112												
@VOLID	001	0006	0092												
@VQ	001	0001	0026	6437	6453	7752	7923	8002	8348	2164	2300	3481	3793*		
@WA37B	001	00FF	1107												
@WSFIT	001	0500	0102												
@WSTBL	001	0503	0103												
@XR	001	0002	0015	4129*	4133	4135	4143	4172	4173	4173	4174	4176	4177	4179	4181

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 31/05/21 PAGE 296

	4186	4188	4188	4194	4196	4197	4198	4198	4213	4245	4274*	4288
	4312*	4328	4328*	4485*	4486	4489	4490	4490	4498	4499	4499	4500
	4500	4503	4511	4626*	4627	4628	4636	4637	4637	4642	4644	4644*
	4654*	4655	4664	4665*	4666	4667	4668	4804	4820	4827	4833	4835
	4846	4848	4849	4849	4855	4865	4894	4908	4910	4910*	4925*	4937
	4938	4939	4982*	4983	4985	4987	5103*	5112	5117	5122	5129	5131
	5131	5132	5143	5146*	5149	5150*	5151	5158	5167	5169	5174	5177
	5189	5191	5210	5220	5241	5361*	5362	5366	5374	5379	5379	5380
	5385	5386	5386	5387	5391	5391	5392	5393	5393	5394	5394	5395
	5395	5396	5396	5397	5397	5398	5409	5410	5422	5425	5439	5442
	5442*	5447*	5590*	5595	5597	5597	5598	5606*	5610	5614	5617	5622
	5629	5634	5647	5651	5653	5654	5656	5658	5663	5669	5685	5690
	5691	5695	5734*	5741	5742	5742	5744	5752	5753	5759	5761	5764
	5898*	5899	5900	5901	5901	5919	5922*	5927	5928	5929*	5930	6041*
	6042	6050	6052	6052	6057	6057	6063	6063	6190*	6191	6195	6198
	6204*	6206	6210	6213	6214	6217	6222	6223	6226	6341*	6342	6344
	6345	6350	6358	6359	6361	6362	6363	6365	6376	6384	6389	6390
	6396	6437	6438	6471	6472	6473	6488	6759*	6763	6764*	6765	6769*
	6770	6811*	6837	6845*	6846	6850	6850*	6851	6862*	6900	6938	6960
	6970*	7026*	7241*	7242	7249*	7250*	7251	7253	7273*	7274	7281*	7287
	7299	7300	7301	7303*	7312*	7313	7314	7315	7370*	7371	7399*	7400
	7401	7405	7410	7410*	7425	7426*	7427	7428	7428*	7429	7434	7435*
	7436	7571*	7572	7572*	7573	7575	7577	7581	7586	7590	7597	7601*
	7602	7618	7626	7627	7633*	7634	7639	7641	7641*	7644	7646	7646
	7647	7647*	7648	7650	7654	7659*	7689	7689	7694	7695	7698	7699
	7700	7701	7703*	7704	7705*	7709*	7710	7718	7718	7719	7723	7724
	7724	7725	7725	7729	7730*	7737	7738	7740	7756	7797	7797	7799
	7824	7835*	7841	7842	7886	7887	7888	7890	7893*	7894	7896	7902
	7902	7903	7907	7908	7915	7916	7917	7923	7925	7932	7933	7935
	7936	7965*	7966	7967	7967	7971*	7977*	7978	7994	7995	7995*	8002
	8009*	8010	8011	8012	8142*	8143	8143*	8144	8146	8148	8152	8156
	8161	8164	8165	8170	8173	8182	8188	8191	8194	8201	8204	8204*
	8206	8209	8212	8226	8230	8230*	8240*	8301	8313	8317	8318	8319
	8320	8323	8329	8342*	8344	8355*	8357	8358	8371	8372	8373	8382*
	8383	8391	8392	8392	8393	8393	8400	8401	8401	8451	8453	8455
	8455	8473	8486*	8496	8498	8499	8502	8506	8507	8508	8512	8515*
	8516	8521	8522*	8529	8538	8542	8544	8552	8557*	8563	8564	8645
	8650	8652	8652*	8656	8658	8660	8662	8663	8664*	8665	8669*	8670
	8679	8679	8680	8680	8681	8681	8686	8687	8689	8689	8690	8690
	8699*	8701	8733	8734*	8737*	8913	8925	8926	8927	8931*	8932	8934*
	8935	8936*	8937	8937*	8938	8939	8939*	8940	8944	8953*	9116	9117
	9119	9123	9125*	9126	9144*	9152	9164	9176*	9184	9216	9226	9228
	9233	9235	9236	9239	9239*	9243	9252	9253	9253*	9260	9260	9261
	9261	9262	9262*	9282*	9291	9294	9325	9369	9391	9395	9414	9463
	9470	9473	9479	9497	9498	9499	9509	9510	9515	9517	9518	9519
	9520	9534	9985*	9986	9987	9987	9998*	0044*	0052	0101	0111*	0120*
	0122	0128*	0165*	0166	0170	0170*	0173*	0178*	0179	0179*	0182*	0183
	0184*	0190*	0191	0191*	0192	0192*	0224*	0228	0247	0501	0501	0502
	0503	0503*	0521	0530	0540	0631	0631*	0632	0703	0714	0716	0718
	0729*	0750	0750*	0751	0753	0755	0755*	0756	0759	0774	0777	0806
	0806*	0807	0895	0927	0934	0936	0943	0948	0952	1006	1006*	1007
	1010	1080	1081	1081*	1086	1092	1094	1096	1101	1133	1133*	1134
	1136	1136*	1137	1147	1153	1158	1159*	1160	1161*	1175	1176	1180*
	1181	1181*	1182	1188	1189*	1194*	1196	1202	1215	1215*	1216	1327
	1340	1352	1360	1379	1390	1418	1419*	1420	1425*	1449	1451	1468
	1468*	1469	1472	1706	1707	1708*	1709	1710*	1726	1729*	1747	1757



## CROSS REFERENCE

SYMBOL   LEN   VALUE   DEFN   REFERENCES   VER 15, MOD 00   31/05/21   PAGE 297

				2036*	2059	2062	2068	2075	2077	2082	2087	2110	2122	2133	2145
				2146	2150	2150*	2163	2164	2164	2168	2169*	2171	2279	2281	2282
				2286	2288	2298	2300	2300	2305	2308	2314	2333	2333	2334	2335
				2340	2343	2345	2356	2365	2389	2586*	2620*	2655*	2869*	2903*	2929*
				2939	3171*	3172	3172*	3182	3184	3190	3197*	3199	3264*	3265	3430
				3434	3434	3436	3438	3441	3442	3452	3473	3475	3480	3480*	3481
				3483*	3491	3492	3496	3508	3509	3513	3517	3521	3522	3532	3574
				3587	3589	3591	3592	3598	3599	3600	3604	3604	3606	3606	3613
				3613	3614	3620	3624	3626	3627	3632	3638	3658	3659	3668	3675
				3676	3684	3688	3689	3691	3692	3693	3693*	3694	3696	3697*	3709
				3734	3739	3752	3775*	3787	3789	3792	3793	3794	3795	3796	3797
				3801	3802	3804	3805								
@ZERO	001	0000	0063	4268	4832	4885	5142	5392	5918	6377	7367	7371	7383	7401	7429
				7644	7648*	7650*	7697	7704*	7732	7818	7827	7914	7980	7989	7994*
				8188	8191	8194	8201	8206	8209	8323	8329	8451	8453	8465	8476
				8498	8656	8731	8742	8933	9153	9330	9342	9386	9402	9528	9533
				0494	0496	0778	0964	1388	1772	2040	2484	2545	2768	2828	3158
				3495	3631	3648	3698	3735	3752						

@4K	001	0010	1061	
B\$ADMK	001	0001	2966	
B\$ADSW	001	159D	2965	
B\$ARMK	001	0001	2951	
B\$ARSW	001	0A45	2950	
B\$BABF	001	1D00	2756	
B\$BCKT	001	1590	2878	
B\$BDPL	001	19E8	2830	
B\$BDSA	001	19EA	2831	
B\$BINO	001	1A6A	2894	
B\$BRLN	001	19F1	2829	
B\$BROP	001	1AF7	2935	
B\$BRVA	001	19EF	2828	
B\$BRVP	001	19EE	2827	
B\$BTAB	001	1996	2826	
B\$CADR	001	1AF9	2936	
B\$CASA	001	0000	2771	
B\$CASC	001	0671	2775	
B\$CASM	001	0608	2773	
B\$CBAS	001	14BB	2901	
B\$CBFA	001	0CBC	2856	
B\$CCGT	001	0600	2781	
B\$CCLS	001	0695	2787	
B\$CCON	001	001F	2854	
B\$CDAT	001	0600	2767	
B\$CDEF	001	0600	2768	
B\$CDIM	001	0673	2769	
B\$CDUM	001	0000	2805	
B\$CEND	001	0600	2803	2804
B\$CEOF	001	0600	2804	
B\$CFOR	001	0600	2776	
B\$CGET	001	06A3	2784	
B\$CGSB	001	0690	2782	
B\$CGTO	001	06B3	2780	
B\$CIFA	001	0600	2778	
B\$CIFC	001	0600	2779	
B\$CIMG	001	0600	2793	
B\$CINP	001	0600	2788	



CROSS REFERENCE																			
SYMBOL	LEN	VALUE	DEFN	REFERENCES													VER 15, MOD 00	31/05/21	PAGE 298
B\$CLTA	001	0000	2770																
B\$CLTC	001	0669	2774																
B\$CLTM	001	0600	2772																
B\$CMAT	001	0600	2794																
B\$CMGT	001	0665	2795																
B\$CMIN	001	06D3	2796																
B\$CMPR	001	069B	2799																
B\$CMPT	001	069B	2798																
B\$CMPU	001	0600	2800																
B\$CMRD	001	06D0	2797																
B\$CNXT	001	0600	2777																
B\$CPCT	001	0CA8	2859																
B\$CPRT	001	0600	2791																
B\$CPRU	001	0600	2792																
B\$CPSE	001	06E7	2801																
B\$CPUT	001	0600	2785																
B\$CPWA	001	0CA6	2930																
B\$CRAD	001	150D	2900																
B\$CRBS	001	1509	2902																
B\$CREA	001	06CF	2789																
B\$CREM	001	0000	2766																
B\$CRMK	001	0001	2978																
B\$CRSR	001	06E3	2790																
B\$CRST	001	06A6	2786																
B\$CRSW	001	0E42	2977																
B\$CRTN	001	06CF	2783																
B\$CSBF	001	0600	2753	2767	2768	2769	2772	2773	2774	2775	2776	2777	2778	2779	2780				
				2781	2782	2783	2784	2785	2786	2787	2788	2789	2790	2791	2792				
				2793	2794	2795	2796	2797	2798	2799	2800	2801	2802	2803	2806				
				2807	2808	2809	2810												
B\$CSCN	001	14B0	2875																
B\$CSMK	001	0007	2981																
B\$CSSW	001	14BC	2980																
B\$CSTP	001	06D6	2802																
B\$CSTR	001	14CC	2899																
B\$CSXA	001	2000	2759																
B\$CTYP	001	0A5F	2853																
B\$CVPD	001	0C5D	2858																
B\$CVPG	001	0CA5	2857																
B\$CWRK	001	F500	2927																
B\$DIST	001	0700	2819																
B\$DLNK	001	1B37	2925																
B\$DL4T	001	1A6B	2896																
B\$DPWA	001	0E46	2931																
B\$DST2	001	073A	2820																
B\$ERMK	001	0007	2954																
B\$ERSW	001	0993	2953																
B\$FACA	001	0E53	2862																
B\$FAIS	001	15AC	2879																
B\$FAIW	001	15A0	2880																
B\$FCON	001	0A46	2852																
B\$FORT	001	1B0E	2921																
B\$FPWA	001	15AC	2932																
B\$FRMK	001	0007	2972																

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 299

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B\$FSC2	001	0E4D	2864	
B\$FSMK	001	0007	2963	
B\$FSSW	001	0E5C	2962	
B\$FSVA	001	0E4F	2865	
B\$FTND	001	1B0B	2923	
B\$FTPT	001	1B0D	2922	
B\$FVME	001	15A2	2884	
B\$FVMP	001	15A4	2885	
B\$FVMS	001	15A6	2886	
B\$FVPE	001	15A8	2881	
B\$FVPP	001	15AA	2882	
B\$FVPS	001	15AC	2883	
B\$GBSW	001	08AF	2956	
B\$GBWK	001	0001	2957	
B\$GETC	001	0867	2833	
B\$GPTR	001	0878	2835	
B\$GTBF	001	1E00	2757	
B\$IFMK	001	0007	2975	
B\$IFSW	001	16E5	2974	
B\$INVT	001	1B38	2915	
B\$KWMK	001	0001	2969	
B\$KWSW	001	159E	2968	
B\$LBAS	001	185E	2906	
B\$LBSV	001	18E7	2904	
B\$LDRP	001	1A00	2754	
B\$LINE	001	07D0	2821	
B\$LIST	001	1853	2888	
B\$LRTN	001	18EB	2905	
B\$LSTR	001	1862	2903	
B\$LTYP	001	18F2	2889	
B\$MATR	001	18F3	2891	
B\$MBMK	001	0007	2990	
B\$MBSW	001	1903	2989	
B\$MFBK	001	1B8F	2917	
B\$MGMK	001	0007	2987	
B\$MGSW	001	18FF	2986	
B\$MPMK	001	0007	2993	
B\$MPSW	001	1981	2992	
B\$MRMK	001	0007	2984	
B\$MRSW	001	0DDE	2983	
B\$NUMC	001	0873	2834	
B\$NXMK	001	0007	2960	
B\$NXSW	001	071D	2959	
B\$PARP	001	0A41	2842	
B\$PBNL	001	0A01	2848	
B\$PCAD	001	0A40	2843	
B\$PCDL	001	09D3	2847	
B\$PCPG	001	0A35	2846	
B\$PECT	001	0A44	2850	
B\$PERC	001	0A39	2849	
B\$PFAE	001	0033	2840	
B\$PFCL	001	009D	2841	
B\$PFNC	001	094E	2838	
B\$PFWP	001	0015	2839	
B\$PNBY	001	0A41	2844	
B\$PPWA	001	0A35	2929	

CROSS REFERENCE																	
SYMBOL	LEN	VALUE	DEFN	REFERENCES	VER 15, MOD 00 31/05/21 PAGE 300												
B\$PRM1	001	1AF3	2933														
B\$PTBF	001	1F00	2758														
B\$PUTC	001	093A	2837														
B\$PVAD	001	0A43	2845														
B\$RMRK	001	1AE6	2898														
B\$RTRN	001	1AF5	2934														
B\$\$SABF	001	1C00	2755														
B\$\$SCAN	001	1514	2877														
B\$\$SCAT	001	13C8	2872														
B\$\$SCON	001	001B	2855														
B\$\$SCVT	001	12E0	2870														
B\$\$SDPL	001	07DA	2823														
B\$\$SFAB	001	0E48	2867														
B\$\$SFNT	001	143C	2873														
B\$\$SLDT	001	109C	2869														
B\$\$SLVT	001	1062	2868														
B\$\$SNAT	001	131A	2871														
B\$\$SPAT	001	07E0	2824														
B\$\$SSTA	001	1BAC	2919														
B\$\$STAS	001	061B	2808														
B\$\$STIF	001	0606	2810														
B\$\$STMA	001	061B	2809														
B\$\$STML	001	0600	2807														
B\$\$STRL	001	0600	2806														
B\$\$SVRB	001	0E46	2866														
B\$\$SYMB	001	0DBC	2861														
B\$TCD2	001	0001	2939														
B\$TLTH	001	0002	2940	2941													
B\$TOD1	001	0000	2938														
B\$TOTB	001	1AF8	2941														
B\$TTAB	001	1AFA	2937	2941													
B\$TYPE	001	0739	2822														
B\$WORK	001	15A0	2926														
B\$ZDBN	001	19F2	2893														
B@ABAS	001	0007	3526														
B@ACD1	001	0001	3523	3524													
B@ACD2	001	0003	3524	3525													
B@AFLG	001	0000	3518														
B@ALLA	001	005C	3343														
B@AMAX	001	0005	3525	3526													
B@BLNK	001	0040	3352	7932 2133	7966 2163	9986	0632	0759	0807	1007	1153	1216	1235	1469	2058		
B@BLSZ	001	0100	3477	3616	3619	3622	3637	3640	7621	7663	7767	8264	8307	8369	8604		
B@BREQ	001	0084	3132														
B@BRHI	001	0088	3133														
B@BRLO	001	0082	3131														
B@BRNE	001	0094	3135														
B@BRNH	001	0098	3136														
B@BRNL	001	0092	3134														
B@CADD	001	0006	3001														
B@CADF	001	0058	3042														
B@CBAS	001	0003	3529														
B@CBNX	001	004A	3035														
B@CBRA	001	0046	3033														

CROSS REFERENCE																
SYMBOL	LEN	VALUE	DEFN	REFERENCES	VER 15, MOD 00 31/05/21 PAGE 301											
B@CBRS	001	004C	3036													
B@CCLS	001	005E	3045													
B@CCMC	001	0042	3031													
B@CCMF	001	0040	3030													
B@CCNT	001	001F	3455	2145												
B@CCSA	001	003E	3029													
B@CDCA	001	006A	3051													
B@CDDL	001	006C	3052													
B@CDIV	001	000C	3004													
B@CDMN	001	0001	3528	3529												
B@CDWA	001	006E	3053													
B@CEOF	001	0070	3054													
B@CEOP	001	0068	3050	1708												
B@CFCI	001	0016	3009													
B@CFN0	001	0012	3007													
B@CFN1	001	0014	3008													
B@CFOR	001	004E	3037													
B@CGET	001	0052	3039	6764												
B@CHAR	001	0000	3468	0052	0521	0530	0540	0632	0703	0714	0716	0718	0751	0753	0756	
				0759	0774	0777	0807	0895	0927	0934	0936	0943	0948	0952	1007	
				1010	1086	1094	1096	1101	1134	1137	1147	1153	1175	1196	1216	
				1327	1340	1352	1360	1379	1390	1449	1451	1469	1472			
B@CHLT	001	0004	3000													
B@CIEX	001	00C5	3428	1264	1276											
B@CIMH	001	0066	3049													
B@CINI	001	0056	3041													
B@CIPI	001	00D7	3431	1260	1272											
B@CIS2	001	00E2	3434	1256	1268											
B@CMF1	001	0018	3010													
B@CMF2	001	001A	3011													
B@CMF3	001	001C	3012													
B@CMA	001	006B	3363	7994	0521											
B@CMPY	001	000A	3003													
B@CMSM	001	001E	3013													
B@CNEG	001	0010	3006													
B@CNXT	001	0050	3038													
B@COLN	001	007A	3365													
B@CPMK	001	00FF	3273	3277	3281	3282	3316									
B@CPRS	001	0060	3046													
B@CPRU	001	0062	3047													
B@CPUT	001	0054	3040													
B@CPWR	001	000E	3005													
B@CRSR	001	005A	3043													
B@CRST	001	005C	3044													
B@CSA1	001	0036	3025													
B@CSA2	001	0038	3026													
B@CSB1	001	003A	3027													
B@CSC1	001	002A	3019													
B@CSD0	001	002E	3021													
B@CSD1	001	0030	3022													
B@CSD2	001	0032	3023													
B@CSF1	001	0022	3015													
B@CSF2	001	0024	3016													
B@CSTA	001	0034	3024													

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 302

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@CSTH	001	0064	3048	
B@CSTX	001	003C	3028	
B@CSUB	001	0008	3002	
B@CSVC	001	0002	2999	
B@CTYP	001	0020	3453	2122
B@CUSC	001	002C	3020	
B@CUSF	001	0026	3017	
B@CVAR	001	005B	3342	
B@DAMK	001	0080	3521	
B@DASA	001	00FF	3282	
B@DASC	001	0040	3286	
B@DASM	001	0038	3284	
B@DCGT	001	0050	3292	
B@DCLS	001	0054	3298	
B@DDAT	001	0024	3278	
B@DDEF	001	0034	3279	
B@DDIM	001	0004	3280	
B@DDUM	001	00FF	3316	
B@DEC0	001	00F0	3411	5612 0943 1010 1340 1411 1472 2068 2087 2314 2343 2380
B@DEC1	001	00F1	3412	4254 4868 4935 5411 6378 2281
B@DEC2	001	00F2	3413	5412 5591
B@DEC3	001	00F3	3414	
B@DEC4	001	00F4	3415	5616
B@DEC5	001	00F5	3416	
B@DEC6	001	00F6	3417	
B@DEC7	001	00F7	3418	
B@DEC8	001	00F8	3419	
B@DEC9	001	00F9	3420	5413 5431
B@DEND	001	0058	3314	3315
B@DEOF	001	0058	3315	
B@DFOR	001	0028	3287	
B@DGET	001	0040	3295	
B@DGSB	001	0020	3293	
B@DGTO	001	0044	3291	
B@DIFA	001	0048	3289	
B@DIFC	001	004C	3290	
B@DIGS	001	007B	3345	
B@DIMG	001	003C	3304	
B@DINP	001	0000	3299	
B@DIVD	001	0061	3362	
B@DLTA	001	00FF	3281	
B@DLTC	001	0040	3285	
B@DLTM	001	0038	3283	
B@DL01	001	0001	3596	3599
B@DL02	001	0003	3599	3602
B@DL03	001	0005	3602	3605
B@DL04	001	0007	3605	3608
B@DL05	001	0009	3608	3611
B@DL06	001	000B	3611	3614
B@DL07	001	0045	3614	3617
B@DL08	001	0145	3617	3620
B@DL09	001	0245	3620	3623
B@DL10	001	0289	3623	3626
B@DL11	001	02C3	3626	3629
B@DL12	001	02FD	3629	3632
B@DL13	001	0337	3632	3635

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 303

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@DL14	001	0371	3635	3638
B@DL15	001	0471	3638	3641
B@DL16	001	0507	3641	
B@DMAT	001	0008	3305	
B@DMGT	001	0044	3306	
B@DMIN	001	0038	3307	
B@DMPR	001	0048	3310	
B@DMPT	001	004C	3309	
B@DMPU	001	0054	3311	
B@DMRD	001	003C	3308	
B@DNXT	001	0044	3288	
B@DPNT	001	004B	3353	7903 0895 1327 1360 2308 2334
B@DPRT	001	002C	3302	
B@DPRU	001	0030	3303	
B@DPSE	001	0050	3312	
B@DPUT	001	0040	3296	
B@DREA	001	000C	3300	
B@DREM	001	00FF	3277	
B@DRSR	001	005C	3301	
B@DRST	001	0050	3297	
B@DRTN	001	005C	3294	
B@DSCY	001	0004	3269	
B@DSIF	001	001C	3318	
B@DSLT	001	0010	3317	
B@DSML	001	0010	3319	
B@DSNS	001	0018	3271	
B@DSS1	001	0000	3270	
B@DSTP	001	0054	3313	
B@DTBN	001	0010	3335	
B@DTB1	001	0050	3334	
B@DTCY	001	0009	3331	
B@DTSN	001	0010	3333	
B@DTS1	001	0040	3332	
B@DTYP	001	0040	3447	6851 7602 7634 7710 8329 8383 1127
B@DURE	001	0020	3165	
B@DVCY	001	0007	3328	3210
B@DVC1	001	0056	3329	
B@DWCY	001	0005	3325	
B@DWT1	001	0003	3326	
B@D1MK	001	0080	3519	
B@D2MK	001	00C0	3520	
B@EOST	001	001E	3341	0052 0530 0540 0751
B@EQUL	001	007E	3367	
B@EXPC	001	00C5	3344	0927 1379
B@FOFL	001	005C	3346	
B@FVAD	001	0001	3531	
B@GETC	001	0001	3470	
B@GETE	001	00FF	3471	
B@GETS	001	0000	3469	
B@GRTR	001	006E	3364	
B@ICON	001	0050	3426	0718 1101
B@LADD	001	0001	3070	
B@LADF	001	0002	3111	
B@LADV	001	0008	3555	3576
B@LBIN	001	0002	3480	3481 3487
B@LBNX	001	0003	3104	

VER 15, MOD 00 31/05/21 PAGE 304

SYMBOL	LEN	VALUE	DEFN	REFERENCES							VER 15, MOD 00		31/05/21		PAGE 304	
B@LBRA	001	0003	3102	7332												
B@LBRC	001	0004	3101													
B@LBRD	001	0003	3103													
B@LBRS	001	0001	3105													
B@LCCA	001	0004	3511													
B@LCCC	001	0001	3063	3101												
B@LCDV	001	0004	3556	3577												
B@LCER	001	0001	3061	3125												
B@LCFN	001	0004	3512													
B@LCLN	001	0002	3066	3117		3118		3125								
B@LCLS	001	0001	3114													
B@LCMC	001	0001	3100													
B@LCMF	001	0001	3099													
B@LCNA	001	0006	3510													
B@LCNN	001	0001	3064	3089	3098	3110	3122	0493	0611	0650						
B@LCOP	001	0001	3060	3068	3069	3070	3071	3072	3073	3074	3075	3076	3077	3078	3079	
				3080	3081	3082	3083	3084	3085	3086	3087	3088	3089	3090	3091	
				3092	3093	3094	3095	3096	3097	3098	3099	3100	3101	3102	3103	
				3104	3105	3106	3107	3108	3109	3110	3111	3112	3113	3114	3115	
				3116	3117	3118	3119	3120	3121	3122	3123	6763	1707			
B@LCRV	001	0013	3554	3574												
B@LCSA	001	0002	3098													
B@LCVA	001	0002	3062	3076	3077	3078	3079	3080	3081	3082	3083	3084	3085	3087	3088	
				3090	3091	3092	3093	3094	3095	3096	3101	3102	3103	3104	3106	
				3107	3108	3120	3121	1726	1757							
B@LCXX	001	0001	3065	3097	3109	3111	3115	3116	0595	0602	0652					
B@LDAT	001	0004	3224													
B@LDCA	001	0003	3120	1782												
B@LDDL	001	0003	3121													
B@LDDM	001	0004	3484													
B@LDEF	001	0003	3225													
B@LDIM	001	0003	3226													
B@LDIN	001	0004	3483	3484		3485										
B@LDIV	001	0001	3073													
B@LDMN	001	0002	3481	3510	3511	3523	3524	3525	3528	3555	3556					
B@LDSN	001	0004	3485	0144	0145	0149	0149	0154	0154	0294	0297	0310				
B@LDWA	001	0002	3122	7332												
B@LELP	001	0010	3553													
B@LEND	001	0003	3253													
B@LEOF	001	0001	3123													
B@LEOP	001	0001	3119													
B@LERC	001	0003	3125													
B@LESP	001	0008	3552													
B@LESS	001	004C	3354													
B@LET\$	001	005B	3374													
B@LET#	001	007B	3375													
B@LET@	001	007C	3376													
B@LETA	001	00C1	3378	0774		1196										
B@LETB	001	00C2	3380													
B@LETC	001	00C3	3381													
B@LETD	001	00C4	3382													
B@LETE	001	00C5	3383													
B@LETF	001	00C6	3384													
B@LETG	001	00C7	3385													
B@LETH	001	00C8	3386													
B@LETI	001	00C9	3387													



## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 305

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@LETJ	001	00D1	3388	
B@LETK	001	00D2	3389	
B@LETL	001	00D3	3390	
B@LETM	001	00D4	3391	
B@LETN	001	00D5	3392	
B@LETO	001	00D6	3393	
B@LETP	001	00D7	3394	
B@LETQ	001	00D8	3395	
B@LETR	001	00D9	3396	
B@LETS	001	00E2	3397	
B@LETT	001	00E3	3398	
B@LETU	001	00E4	3399	
B@LETV	001	00E5	3400	
B@LETW	001	00E6	3401	
B@LETX	001	00E7	3402	
B@LETY	001	00E8	3403	
B@LETZ	001	00E9	3404	
B@LEXP	001	0008	3443	
B@LFCI	001	0003	3078	7333
B@LFNA	001	0002	3557	3578
B@LFN0	001	0003	3076	
B@LFN1	001	0003	3077	
B@LFOR	001	0003	3106	
B@LFRT	001	0004	3498	3499
B@LGET	001	0003	3108	
B@LGSB	001	0005	3232	
B@LGTO	001	0004	3231	
B@LHLT	001	0001	3069	
B@LIEX	001	0002	3429	
B@LIFN	001	0003	3492	
B@LILP	001	0009	3551	3569 3570 3571
B@LIMG	001	0001	3243	
B@LIMH	001	0003	3118	
B@LINI	001	0002	3110	
B@LINP	001	0005	3238	
B@LIPI	001	0003	3432	
B@LISP	001	0005	3550	3558 3564 3565 3566
B@LIS2	001	0005	3435	
B@LIVT	001	0001	3508	
B@LKCL	001	0005	3237	
B@LKFR	001	0003	3228	
B@LKGT	001	0003	3234	
B@LKIF	001	0002	3230	
B@LKON	001	0002	3263	
B@LKPT	001	0003	3235	
B@LKPU	001	000A	3242	
B@LKRR	001	0007	3240	
B@LKRT	001	0005	3236	
B@LKTO	001	0002	3257	
B@LLET	001	0003	3227	
B@LL01	001	0002	3595	3596
B@LL02	001	0002	3598	3599
B@LL03	001	0002	3601	3602
B@LL04	001	0002	3604	3605
B@LL05	001	0002	3607	3608
B@LL06	001	0002	3610	3611

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 306

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@LL07	001	003A	3613	3614
B@LL08	001	0100	3616	3617
B@LL09	001	0100	3619	3620
B@LL10	001	0044	3622	3623
B@LL11	001	003A	3625	3626
B@LL12	001	003A	3628	3629
B@LL13	001	003A	3631	3632
B@LL14	001	003A	3634	3635
B@LL15	001	0100	3637	3638
B@LL16	001	0096	3640	3641
B@LMAT	001	0003	3244	
B@LMF1	001	0003	3079	
B@LMF2	001	0003	3080	
B@LMF3	001	0003	3081	
B@LMGT	001	0006	3245	
B@LMIN	001	0008	3246	
B@LMPR	001	0008	3249	
B@LMPT	001	0006	3248	
B@LMPU	001	000D	3250	
B@LMPY	001	0001	3072	
B@LMRD	001	0007	3247	
B@LMSM	001	0003	3082	
B@LNEG	001	0001	3075	
B@LNEX	001	0004	3229	
B@LNXT	001	0003	3107	
B@LPAR	001	004D	3355	
B@LPRS	001	0002	3115	
B@LPRT	001	0005	3241	
B@LPRU	001	0002	3116	
B@LPSE	001	0005	3251	
B@LPUT	001	0002	3109	
B@LPWR	001	0001	3074	
B@LREA	001	0004	3239	
B@LREM	001	0003	3223	
B@LRSR	001	0001	3112	
B@LRST	001	0001	3113	
B@LRTN	001	0006	3233	
B@LSA1	001	0003	3094	
B@LSA2	001	0003	3095	
B@LSB1	001	0003	3096	
B@LSC1	001	0003	3088	
B@LSDF	001	0004	3478	
B@LSD0	001	0003	3090	
B@LSD1	001	0003	3091	
B@LSD2	001	0003	3092	
B@LSF1	001	0003	3084	
B@LSF2	001	0003	3085	
B@LSKW	001	0002	3494	
B@LSNO	001	0002	3487	0143 0293
B@LSPT	001	0003	3502	3505
B@LSTA	001	0003	3093	
B@LSTC	001	0003	3087	
B@LSTE	001	0004	3258	
B@LSTF	001	0003	3083	
B@LSTH	001	0003	3117	
B@LSTP	001	0004	3252	

CROSS REFERENCE																		
SYMBOL	LEN	VALUE	DEFN	REFERENCES												VER 15, MOD 00	31/05/21	PAGE 307
B@LSTX	001	0002	3097															
B@LSUB	001	0001	3071															
B@LSVC	001	0001	3068															
B@LTHN	001	0004	3259															
B@LTYP	001	0001	3488															
B@LUFN	001	0002	3495															
B@LUSC	001	0002	3089															
B@LUSF	001	0001	3086															
B@LVPG	001	0100	3582	3585	4098	4243	4377	4789	4980	4992	4993	5098	5356	5583	5729			
				5797	6185	6236	6237	6238	6239	6336	6421	6507	6508	6753	7233			
				7354	7563	7680	7791	7875	7880	8135	8287	8435	8621	9986*	9987			
				9987	9987*	1680												
B@MINS	001	0060	3361	7917	7935	0716	0936	1096	1098	1267	1271	1275	1451	2061	2363			
B@MULT	001	005C	3358															
B@NAAR	001	001D	3546	3576	3628													
B@NCAR	001	001D	3547	3577	3631													
B@NCRV	001	001D	3545	3574	3625													
B@NDGT	001	000A	3538	3544														
B@NEQL	001	007F	3368															
B@NFRT	001	000A	3497	3499														
B@NICN	001	0006	3540	3542														
B@NIEL	001	0007	3542	3558	3564	3569												
B@NIFN	001	0018	3491															
B@NIVR	001	0001	3541	3542														
B@NIVT	001	0057	3507															
B@NLDV	001	0122	3544	3566	3571	3622												
B@NLRV	001	001D	3543	3565	3570	3613												
B@NLTR	001	001D	3537	3543	3544	3545	3546	3547	3548									
B@NSKW	001	0004	3493															
B@NSPT	001	0028	3501															
B@NUFN	001	001D	3548	3578	3634													
B@NVPG	001	0100	3581	3585														
B@NXHI	001	00E3	3462	1026														
B@NXLO	001	001E	3461	4655	4848	5598	1027	1413										
B@NXZR	001	0080	3460	3461	3462	4233	4235	4486	4498	4518	4629	4652	4804	4938	4985			
				5152	5167	5201	5258	5595	5610	5622	5634	5717	5721	5723	5782			
				5784	5787	5789	5792	5909	5939	6195	6206	6342	6350	6415	8024			
				1025	1307	2075	2077	2199	2286	2288	2297	2364	2405					
B@PLUS	001	004E	3356	0714	0934	1093	1094	1255	1259	1263	1449							
B@POWR	001	005A	3357															
B@PREC	001	0020	3449	7719	7723	8391	8400											
B@PROD	001	0023	3558															
B@PRPL	001	0002	3145	7605														
B@PRPN	001	0001	3144															
B@PRPR	001	0004	3147	2201														
B@PRPS	001	0003	3146															
B@PRRC	001	0007	3150	7599														
B@PRRL	001	0008	3151	2200	2201													
B@PRSL	001	0005	3148	6938	2042													
B@PRSS	001	0006	3149															
B@PTAB	001	0000	3503															
B@PTAD	001	0001	3504														</	

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 308

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@PUI1	001	0004	3156	
B@PUI2	001	0005	3157	
B@PUNL	001	0002	3159	
B@PUNS	001	0003	3160	
B@PUTM	001	0010	3164	
B@RPAR	001	005D	3359	
B@SADV	001	00E8	3576	3579
B@SAVL	001	0B76	3572	3589
B@SAVS	001	065E	3567	3588
B@SCDV	001	0074	3577	3579
B@SCLN	001	005E	3360	
B@SCRV	001	0227	3574	3588 3589
B@SDMK	001	0080	3489	8535
B@SEXP	001	0004	3442	
B@SFAT	001	0196	3579	3588 3589 3640
B@SFNA	001	003A	3578	3579
B@SFRT	001	0028	3499	
B@SIEL	001	003F	3569	3572
B@SIES	001	0023	3564	3567
B@SIGN	001	0010	3451	
B@SLDL	001	0A32	3571	3572
B@SLDS	001	05AA	3566	3567
B@SLVL	001	0105	3570	3572
B@SLVS	001	0091	3565	3567
B@SQUO	001	007D	3366	7639 7644 7648 7650 7896 8201 8209 8212 0703 0753 0756 1086 1134 1137
B@STAT	001	0000	3441	
B@TASA	001	0012	3176	
B@TASC	001	001E	3182	
B@TASM	001	0018	3178	
B@TASS	001	007B	3183	
B@TCGT	001	0030	3191	
B@TCLS	001	0042	3197	
B@TDAT	001	0006	3172	
B@TDEF	001	0009	3173	
B@TDIM	001	000C	3174	
B@TDUM	001	0078	3215	
B@TEND	001	0072	3213	
B@TEOF	001	0075	3214	
B@TFOR	001	0021	3185	
B@TGET	001	0039	3194	
B@TGSB	001	0033	3192	
B@TGTO	001	002D	3190	
B@TIFA	001	0027	3187	
B@TIFC	001	002A	3188	
B@TIFS	001	007D	3189	
B@TIMG	001	0054	3203	
B@TINP	001	0045	3198	
B@TLTA	001	000F	3175	
B@TLTC	001	001B	3179	
B@TLTM	001	0015	3177	
B@TLTS	001	0079	3180	
B@TMAS	001	007C	3184	
B@TMAT	001	0057	3204	
B@TMGT	001	005A	3205	
B@TMIN	001	005D	3206	

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 309

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@TMLS	001	007A	3181	
B@TMPR	001	0066	3209	
B@TMPT	001	0063	3208	
B@TMPU	001	0069	3210	
B@TMRD	001	0060	3207	
B@TNXT	001	0024	3186	
B@TPRT	001	004E	3201	
B@TPRU	001	0051	3202	
B@TPSE	001	006C	3211	
B@TPUT	001	003C	3195	
B@TRAC	001	0080	3445	
B@TREA	001	0048	3199	
B@TREM	001	0003	3171	
B@TRSR	001	004B	3200	
B@TRST	001	003F	3196	
B@TRTN	001	0036	3193	
B@TSTP	001	006F	3212	
B@VMC1	001	0056	3584	
B@VMLB	001	F0CD	3589	
B@VMSB	001	F5E5	3588	
B@VMSZ	001	0000	3585	3587 3588 3589
B@VMTB	001	0000	3587	
B@ZNEG	001	00D0	3458	1369
B@ZPOS	001	00F0	3457	4135 4628 4820 4983 5158 5191 5362 5617 5928 7933 7936 0957 0958 1369 1414 2059 2062 2340 2345
BFPCAR	001	4FFB	3714	3717
BFPCRO	001	4FFF	3716	3696*
BUFADR	002	4DD9	3546	3438* 3480 3591* 3693
BUFRWK	002	4DDE	3550	9517* 3592* 3604* 3626
CBFAD1	001	0C70	5889	5893
CBFEXP	001	0002	5938	5920 5921 5939
CBFPZD	004	0C70	5898	
CBFSFT	002	0CB1	5939	5920
CBF100	004	0C97	5922	5919* 5920* 5921*
CBF900	004	0CAC	5934	5910
CCZAD1	001	04AD	4617	4621
CCZDC1	001	04FB	4679	4645
CCZDFP	004	04AD	4626	
CCZEXP	001	04FA	4674	4629* 4645* 4652 4668
CCZONE	001	0001	4678	4644
CCZSGN	001	04F9	4673	4627* 4667
CCZ020	003	04C2	4642	4653
CCZ100	005	04DF	4664	4643
CCZ900	004	04F5	4669	4656
CDBACC	001	0004	6068	6052* 6063
CDBADD	001	0003	6069	6052 6052 6057 6057 6063
CDBAD1	001	0CB2	6032	6036
CDBNZD	004	0CB2	6041	
CDBONE	001	0CDA	6073	6042
CDB010	003	0CBD	6050	6049* 6058 6058* 6059
CDB100	004	0CC7	6057	6051
CENAD1	001	0470	4476	4480
CENXZD	004	0470	4485	
CENZRO	001	04AC	4518	4511
CEN100	003	0487	4498	4488
CEN150	003	0498	4503	

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 310

SYMBOL	LEN	VALUE	DEFN	REFERENCES
CEN200	004	049E	4511	4487
CEN900	004	04A8	4514	4491 4504
DENTRY	001	0025	3567	3440
DERROR	003	0088	3566	3490
DFKACK	001	0010	9092	9233
DFKATA	001	261C	9055	9124 9268*
DFKATC	001	2733	9220	9243 9252
DFKBLE	002	2617	9051	8940* 9125
DFKBSP	001	0016	9086	9199
DFKBS2	001	2600	9035	8920 8922 8931 8934 9076 9194 9256* 9264* 9268*
DFKBS3	001	2700	9195	8913 8942 9077 9123* 9152* 9280
DFKCNT	001	2624	9063	9143*
DFKC01	002	2621	9058	9060 9225 9241 9282
DFKDIO	001	0065	8947	8918
DFKDLP	001	2696	9148	9114 9242
DFKDTK	001	0040	9099	9120
DFKEMS	001	0002	9091	9208
DFKENB	001	0012	9096	9110
DFKENT	001	2653	9113	8934
DFKERA	001	2789	9259	9204
DFKERS	001	0003	9088	9203
DFKEUD	001	001D	9094	
DFKEXL	001	0019	9098	9036
DFKEYN	001	2500	8915	8912 8920 8937 8942 8947 8957 9034 9076 9077
DFKJAR	002	2615	9050	8932* 8945
DFKIET	002	2619	9052	9103
DFKIME	002	262C	9068	9224* 9225*
DFKIRK	001	2634	9073	9041
DFKIST	002	2621	9060	9293
DFKKIX	001	0011	9093	9235
DFKLLA	001	25F9	9016	9024
DFKLMG	002	2628	9066	8925* 8953 9237 9256 9264
DFKLNK	001	0039	9024	9268
DFKLOK	001	0018	9095	9104 9223
DFKMCT	002	262E	9069	9224
DFKMSD	002	27B1	9280	9261
DFKNAB	001	264D	9109	9106
DFKNPS	002	261F	9057	8928* 8929* 8930 8933* 9141* 9143 9150 9153*
DFKNSK	001	261D	9056	9040* 9041 9115 9118 9120 9197 9199 9201 9203 9206 9208 9210
				9221 9227* 9251
DFKNTR	001	2603	9038	8931
DFKPG2	002	263A	9076	8951
DFKPG3	002	263C	9077	8949
DFKPL1	001	2770	9245	9233* 9235* 9239
DFKPL2	001	27A3	9271	9253
DFKPL3	001	27A5	9273	9260* 9261* 9262
DFKPPL	001	2623	9061	9065 9144 9150* 9183
DFKPRT	001	26AC	9162	9145 9240 9254 9263 9283
DFKP10	001	26BD	9168	8923* 8924* 9165 9170
DFKP20	002	26BF	9169	9164*
DFKRET	002	2630	9070	8938* 9105* 9111
DFKRKY	001	0011	9090	9072 9197
DFKRMG	002	262A	9067	8927* 8930* 9183
DFKROR	001	27BB	9288	9116
DFKROS	002	2632	9071	8935* 9039
DFKRTN	001	0015	9087	9201

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 311

SYMBOL	LEN	VALUE	DEFN	REFERENCES
DFKRT1	001	2683	9140	9127 9217
DFKSG1	001	0007	9279	9275
DFKSPA	001	27A9	9277	9279 9280
DFKSPB	001	279D	9267	9207
DFKSPC	001	274D	9232	9200
DFKSTN	001	0040	9089	9206
DFKTAB	001	2626	9065	8926* 9151 9152 9237 9241* 9256* 9264*
DFKTBL	001	0005	9085	9210
DFKTST	001	25C0	8958	8939 9024
DFKULK	001	26DD	9181	9122 9205
DFKXDP	001	001C	9097	9108 9185
DFKXIT	002	2638	9075	9105
DFKXRS	001	264A	9107	9121 9128 9211 9222 9265 9284 9299
DFK001	002	2636	9074	8944* 9176
DFK100	001	0001	9084	9141
DFK120	004	2565	8948	8917* 8918* 8937 8939 9075
DFK140	005	2569	8949	9075
DFK160	004	257E	8954	
DFK180	003	2600	9036	9042
DFK200	003	263D	9103	9198 9257
DFK220	004	2671	9124	9269
DFK240	005	2678	9126	9124* 9151*
DFK260	004	2699	9150	9146
DFK280	004	26A8	9154	9142* 9149*
DFK300	004	26C8	9172	9166
DFK320	003	26D6	9176	9171
DFK340	004	26D9	9177	9163*
DFK350	004	26EA	9186	9182*
DFK360	001	2700	9196	9119 9228
DFK380	004	2740	9225	9226
DFK400	003	2750	9234	9123* 9216* 9236*
DFK420	004	2759	9237	9234
DFK440	003	276D	9243	9238
DFK460	003	2772	9251	9184
DFK480	003	2778	9253	9202
DFK500	004	277E	9255	9152* 9291
DFK520	003	27B2	9282	9209
DFK540	003	27BB	9289	9117* 9294*
DFPAPC	005	27C5	9293	9289
DFPASE	002	28DD	9422	9364* 9365* 9367 3694* 3695*
DFPASY	001	2800	9420	9314 9384* 9432 9436 9445 9462 3429 3586 3733 3756* 3786
DFPCFD	002	29D5	9540	9497* 9498* 9499
DFPCHK	002	28EB	9432	9365
DFPDSV	001	5300	3732	
DFPENT	004	28F4	9437	9454 9495 3741*
DFPEOR	004	5311	3741	3736
DFPERC	001	4DF3	3573	3532
DFPERR	001	28EE	9435	9383
DFPETN	004	29DD	9544	
DFPEXT	001	28E9	9431	9354
DFPGCT	001	29D3	9539	9509
DFPIOR	001	0000	9546	9488*
DFPIST	001	29D7	9542	
DFPITE	001	28F5	9438	9328 9330* 9338 9339* 9340* 9342* 9344 9348* 9440 9452 9495* 3735
				3739* 3741 3744
DFPITE	002	28E7	9447	9408



## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 312

SYMBOL	LEN	VALUE	DEFN	REFERENCES
DFPLBU	002	29D2	9538	9510
DFPMCK	001	2939	9487	9480
DFPNDX	001	2900	9464	9413 9463
DFPOFF	001	28E3	9428	9388
DFPOGE	001	29DD	9545	9473
DFPORK	002	28E5	9429	9347* 9348 9453
DFPPCF	001	28DE	9423	9331 9344* 9346* 9347 9350 9351* 9354* 9355 9357* 9359* 9361* 9363*
				9432 9456 3744*
DFPPCH	002	28FD	9445	9316 9321 9401
DFPPCO	001	28E2	9427	3627* 3696
DFPPOS	001	4DE4	3556	3632* 3638 3658* 3659 3668 3675 3676* 3684* 3688* 3689
DFPRCK	001	28A5	9382	
DFPRCL	001	0002	9421	9383 9434
DFPRCT	002	28ED	9434	9383* 9488* 9493*
DFPRES	001	4DDC	3549	3598* 3606 3613* 3620 3801 3802
DFPRNT	001	2800	9315	9413 9420 3542 3572 3631* 3667*
DFPRPE	001	28D3	9411	9393 3542 3572
DFPRSN	002	29D9	9543	9470* 9479 9515
DFPSCK	001	2932	9481	
DFPSC2	001	2948	9492	9486
DFPSYC	001	28F9	9442	9436 9490* 9500* 9501 9502* 9504* 3743*
DFPULK	001	5339	3754	3737
DFPVCK	001	0004	9547	9515
DFPWITH	002	4DDB	3548	3599* 3600* 3604 3606 3613 3614
DFPX39	001	0039	9446	9322
DFPYCD	002	28F0	9436	9498
DFPYCT	001	0001	9448	9493*
DFP001	002	28E7	9430	9333 9351 9363 9447 9455 9478 9488 9493 9502
DFP100	004	2805	9317	9323 9403 9445
DFP101	004	280E	9319	9318*
DFP102	005	2812	9321	9404
DFP105	002	281F	9324	3757
DFP115	001	2823	9326	3746 3748
DFP120	003	283D	9338	9329
DFP140	004	2853	9344	9341
DFP160	005	2862	9350	9343
DFP180	003	2872	9355	9332 9353
DFP200	005	2878	9357	
DFP220	006	2888	9362	9360
DFP240	003	2892	9364	9334 9356 3499
DFP250	001	2899	9366	3699
DFP260	003	289C	9368	9386* 9522 3631* 3667*
DFP270	003	289F	9369	9370 9372 3498* 3531* 3621*
DFP280	003	28A2	9373	9384* 3444 3756*
DFP300	003	28CA	9404	9536 3535
DFP320	003	28AC	9385	9373
DFP330	003	28BB	9395	9396 9398 9400 9509* 3433* 3440* 3490* 3530*
DFP333	001	28BA	9394	3442* 3532*
DFP335	003	28B8	9389	9390 9392
DFP340	003	28B2	9387	9409
DFP360	003	28CD	9408	9387
DFP378	001	2911	9469	9466
DFP380	005	2927	9478	9472
DFP400	004	29B2	9527	9489 9494
DFP420	003	2953	9497	9491
DFP440	004	296F	9505	9503

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 313

SYMBOL	LEN	VALUE	DEFN	REFERENCES
DFP480	001	29CE	9535	9530
DLFBPT	001	4DDF	3551	9518* 9519 3624* 3691
DLFCAR	001	00FB	3717	3695
DLFDSV	004	28F4	9454	3788* 3791 3793 3802*
DLFEOR	001	4DF6	3575	3442
DLFIST	001	28F5	9452	3453 3468 3469 3487 3791* 3792 3800* 3801*
DLFMAR	002	4DE1	3553	
DLFORK	002	28E5	9453	9512* 9517 3648* 3649* 3650 3650* 3651 3651* 3652 3652* 3653 3653*
				3654 3654* 3659 3692* 3697
DLFPCF	001	28DE	9456	3497* 3533* 3637* 3677* 3686* 3689*
DLFPCH	002	4DED	3562	3473
DLFPC1	002	4DEF	3563	3521
DLFPRT	001	4D00	3432	9369 9391 9399 9517* 9518* 9519 9520* 9534 9539 3430 3562 3563
				3566 3567 3587 3734 3787
DLFRPE	001	4DCD	3538	9391 3574
DLFRTN	001	0001	3568	3789 3804
DLFRTY	002	4DE3	3554	3509
DLFSWC	001	4DF0	3564	3789 3796* 3804*
DLFVD1	002	4DD7	3545	3436 3589
DLFVD2	002	4DEB	3558	3491
DLFX4E	001	004E	3559	3759
DLFX53	001	0053	3560	3761
DLF001	002	28E7	9455	3636 3665 3688 3788 3795
DLF050	001	4D18	3439	
DLF100	001	4D25	3451	9399 3517 3567 3805
DLF140	001	4D3F	3464	3461
DLF143	004	4D4E	3474	3522
DLF145	004	4D57	3476	3475*
DLF146	003	4D5B	3480	3797
DLF150	005	4D5E	3481	3792* 3793* 3794* 3795*
DLF155	004	4D63	3483	3434*
DLF160	001	4D6D	3489	3496
DLF165	005	4D70	3491	
DLF170	001	4D78	3494	3454
DLF175	001	4DB9	3529	3488 3563
DLF350	003	4D88	3504	9520* 9539 3505 3507 3508* 3566
DLF355	005	4D8E	3509	
DLF360	003	4DA8	3517	3441* 3452* 3504 3518 3520
DLF375	003	4DB0	3522	
DLF400	001	4D93	3510	3492 3562
DLF425	004	4DA4	3515	3513*
DLF450	001	29AF	9521	9506 9516
DLF500	001	4E25	3603	3553
DLF525	001	4E29	3605	3601
DLF550	001	4E3C	3619	3607
DLF600	001	4E44	3623	3615
DLF700	001	4E49	3625	3554
DLF800	001	4EC7	3683	3670
DLF900	001	4ECC	3685	3660
DLF920	001	4ECF	3687	3678
DLF950	001	4ED7	3690	3639 3669
DLF960	004	539F	3792	3790
DLTABL	001	0090	3561	3686
DLTABR	001	00A0	3565	3677
FBSADA	008	009F	6503	6428
FBSATA	008	129F	6497	6503

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 314

SYMBOL	LEN	VALUE	DEFN	REFERENCES
FBSATN	004	1100	6341	6503
FBSAT1	008	1188	6414	6389
FBSBN1	001	1297	6493	6474
FBSINS	005	128E	6491	6477
FBSINZ	004	1285	6488	6427
FBSLNR	001	0009	6405	6376 6384 6448 6453 6491
FBSLNT	001	128D	6490	6470
FBSLNW	001	000B	6406	6376 6378* 6438 6438 6449 6449 6452 6452 6471 6471 6472 6473 6482 6483
FBSLST	001	12B6	6502	6491
FBSMDS	004	1296	6492	6469
FBSMDZ	004	128C	6489	6436
FBSONE	007	1190	6416	6358 6363
FBSRND	001	1180	6413	6390
FBSRRR	001	0614	6407	6384 6448* 6453* 6491*
FBSSGN	001	117F	6412	6344* 6396
FBSWWW	001	061F	6408	6376* 6378* 6438 6449* 6452* 6471
FBSZER	001	0005	6404	6376 6376
FBSZZZ	011	1278	6482	6437* 6449 6452 6488*
FBS10Y	011	1283	6483	6472* 6473
FBS100	003	1111	6350	
FBS110	003	111D	6357	6351
FBS190	004	1148	6370	6353
FBS200	005	114E	6376	
FBS400	004	120A	6436	6478
FBS405	004	120E	6437	6427* 6429 6436* 6450
FBS420	005	1212	6438	6454 6476
FBS425	005	1224	6448	6428* 6470*
FBS430	005	1229	6449	
FBS440	005	1231	6452	6446
FBS450	005	1236	6453	6447 6469* 6477*
FBS600	004	123E	6458	6439
FBS800	003	1166	6388	6352* 6357*
FBS810	004	116D	6390	6388
FBS900	004	117B	6400	6343
FGSBN1	001	05CF	4954	4836 4857 4909 4936
FGSEVP	004	0500	4797	
FGSFVE	001	05D0	4955	4926
FGSINL	001	0005	4949	4884 4916 4956
FGSINS	006	05F5	4965	4884
FGSITN	001	05FC	4970	4885* 4906 4909*
FGSMNN	010	05E9	4959	4846
FGSMOD	005	05D5	4956	4916
FGSNNL	001	000A	4950	4957 4959 4961
FGSNNN	010	05DF	4957	4827
FGSONE	001	0001	4947	4827 4835 4855 4865 4867* 4894 4908 4961
FGSSFZ	002	0619	4990	4987
FGSTEN	011	05F4	4961	4833 4835 4855
FGSTHR	001	0003	4948	4827 4833 4835 4846 4855 4950
FGSXM1	001	05FB	4969	4832* 4836* 4854* 4857* 4936* 4939
FGS001	004	0600	4982	
FGS004	004	0614	4988	4984 4986
FGS005	004	050C	4806	4828
FGS010	004	0513	4814	4805
FGS100	004	0529	4833	4837
FGS110	004	053B	4846	4821

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 315

SYMBOL	LEN	VALUE	DEFN	REFERENCES
FGS115	003	054C	4854	4847
FGS120	004	054F	4855	4858
FGS210	005	055D	4865	4834 4856
FGS220	005	057D	4894	4905 4917
FGS250	006	0585	4901	4884* 4916*
FGS260	003	058E	4906	4895
FGS300	004	05A7	4925	4907
FGS305	005	05BF	4937	4928
FGS900	004	05CB	4943	4807 4850
FKSADD	001	0002	4224	4194
FKSARG	008	037C	4347	4253* 4254* 4258* 4279* 4321* 4332 4333*
FKSCNT	008	0093	4369	4894 4908
FKSCNV	008	02A3	4229	4107* 4118* 4213
FKSCON	008	0393	4359	4288 4369
FKSDCR	001	02BC	4238	4177 4186
FKSINC	001	038A	4354	4301
FKSINS	006	036F	4343	4267
FKSINT	001	0005	4338	4267
FKSITN	001	0384	4349	4268* 4301* 4305
FKSLGT	004	0200	4107	4117 4126
FKSLOG	003	0219	4127	
FKSLTW	004	020B	4118	
FKSMDY	005	0389	4353	4327
FKSMOD	001	0005	4339	4327 4353
FKSONE	001	0001	4223	4173 4173 4176 4179 4181 4186* 4188* 4194* 4196 4197 4198 4251* 4252 4288 4310 4333*
FKSRND	001	038B	4355	4310
FKSSFT	001	0002	4225	4197*
FKSSHT	007	0383	4348	4332* 4333
FKSTEN	007	02AB	4234	4107
FKSTNE	008	02BB	4237	4176 4181
FKSTWO	007	02B3	4236	4118
FKS010	003	0212	4120	4109
FKS020	004	021F	4129	4121
FKS025	004	022F	4137	4108* 4119* 4128* 4134
FKS030	005	0236	4143	4136
FKS090	004	0300	4251	4369
FKS095	004	0321	4274	4273*
FKS100	005	0325	4279	4297 4334
FKS120	006	0332	4296	4267* 4327*
FKS150	004	033B	4301	4280
FKS175	005	0358	4321	4306
FKS205	003	024E	4174	4178
FKS210	003	025F	4179	4175 4187
FKS220	004	0270	4188	4180
FKS600	003	028D	4212	4120* 4127*
FKS700	004	0298	4219	4138 4212
FNBBN1	001	08EC	5257	5145
FNBCNT	001	08E0	5250	5142* 5145* 5151* 5152 5201
FNBDCl	001	08EE	5260	5175
FNBDGT	001	08E1	5251	5149* 5175* 5192
FNBFPl	001	08EE	5259	5132
FNBMK1	001	0002	5246	5132
FNBMN1	002	08EB	5256	5146
FNBPWR	004	0800	5103	
FNBSTR	008	08E9	5252	5210* 5220

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 316

SYMBOL	LEN	VALUE	DEFN	REFERENCES
FNB005	003	0810	5122	
FNB010	003	081D	5129	5113
FNB030	003	082E	5142	5130
FNB200	003	0831	5143	5147
FNB250	004	0841	5149	5144
FNB275	004	0859	5160	5118
FNB300	003	0860	5167	5153
FNB350	004	0871	5175	5181
FNB400	003	088B	5189	5168 5170
FNB500	003	08A6	5209	5159 5190 5193 5202
FNB800	004	08A9	5210	5208
FNB880	003	08D6	5240	5203* 5209* 5234
FNB900	004	08DC	5242	5123 5125 5133 5161 5176 5180 5226 5236 5240
FRBACC	001	0001	5454	5425*
FRBBN1	001	09B4	5472	5385
FRBDC1	001	09B5	5473	5425
FRBEVN	001	0001	5453	5379*
FRBEXP	001	0002	5456	5393 5394 5395
FRBFC1	009	09AC	5462	5412* 5424
FRBFC2	007	09B3	5468	5407* 5408 5408* 5409 5410 5413* 5431 5440 5441 5441*
FRBLNG	001	0001	5457	5422* 5439*
FRBNRM	001	09B6	5474	5398
FRBONE	001	0001	5452	5391*
FRBSQR	004	0900	5361	
FRBSUB	009	09A3	5461	5411* 5422 5424* 5439 5440*
FRBTWO	001	0002	5455	
FRB005	003	0911	5366	5363
FRB010	003	0917	5374	
FRB020	004	0927	5385	5375
FRB030	004	0932	5391	5381
FRB100	004	0969	5422	5426 5443
FRB150	003	097B	5431	5423
FRB400	003	097E	5432	
FRB850	004	0993	5447	5432
FRB900	004	0997	5448	5365 5367
FSSADD	001	0003	5708	5647
FSSCOF	007	0B70	5783	5745
FSSCOS	004	0A00	5590	5584
FSSDCO	001	0B67	5777	5751
FSSEQ8	001	0001	5706	
FSSFP1	007	0AC8	5718	5629
FSSHLF	007	0AD6	5722	5669 5685
FSSINP	008	0B66	5773	5741* 5764
FSSINT	001	0003	5709	5651 5653 5653 5654 5656 5656 5719 5720
FSSLOP	001	0B5E	5772	5740* 5762*
FSSMDY	001	0AD8	5724	5690
FSSMN1	001	0B68	5778	5762
FSSMOD	001	0002	5707	5690
FSSOCT	001	0AC0	5713	5591* 5612* 5616* 5647 5658* 5686 5688 5693 5699
FSSONE	001	0001	5705	5690*
FSSRST	008	0B5D	5771	5745* 5753 5761*
FSSSIN	004	0A1A	5606	
FSSSQD	008	0B55	5770	5744* 5752
FSS008	003	0ACE	5720	5654 5656
FSS050	003	0A14	5599	5596
FSS064	003	0ACB	5719	5651 5653

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 317

SYMBOL	LEN	VALUE	DEFN	REFERENCES
FSS100	003	0A33	5617	5600
FSS150	003	0A36	5622	5615
FSS160	004	0A3C	5624	5599* 5613*
FSS200	004	0A43	5629	5623
FSS205	003	0A4B	5634	
FSS225	004	0A5B	5651	
FSS230	004	0A66	5654	5652 5657
FSS260	004	0A74	5658	5655
FSS300	004	0A81	5669	5635
FSS360	003	0A9D	5691	5700
FSS370	004	0AA0	5692	5701
FSS380	003	0AA4	5693	5689
FSS400	004	0AAD	5696	5694
FSS425	004	0AB3	5698	5611 5625
FSS450	003	0AB7	5699	5687
FSS900	004	0B00	5734	5697
FSS905	004	0B17	5745	5746
FSS910	004	0B1F	5751	5763
FSS920	004	0B2F	5759	5746* 5751*
FWSCOT	004	0D00	6190	
FWSLRG	001	0003	6230	6195 6206
FWSPCH	120	0DFB	6231	
FWSSAV	008	0D27	6200	6198* 6210* 6214 6223
FWSTAN	004	0D28	6204	
FWS005	003	0D10	6195	6192
FWS007	003	0D2F	6206	
FWS009	004	0D35	6208	6193* 6194 6196* 6197 6205*
FWS030	004	0D3C	6210	6207
FWS040	004	0D62	6220	6199
FWS900	004	0D80	6228	6209
FZRBAT	001	336D	1788	1708
FZRBN1	001	336B	1781	1771
FZRDCA	001	332F	1722	1791
FZRDDL	001	3349	1743	1792
FZREAD	001	3300	1685	1790 1791 1792
FZREOP	001	335F	1766	1790
FZRLDA	001	336C	1782	1734
FZR010	004	3307	1692	
FZR020	005	330F	1699	1688 1758 1773
FZR030	002	3319	1702	1699*
FZR040	003	331A	1706	
FZR050	004	3324	1709	1707*
FZR060	004	3328	1710	1706*
FZR070	003	332C	1714	1709*
FZR080	005	332F	1726	
FZR090	005	3340	1734	
FZR100	003	3349	1747	
FZR110	004	334F	1752	
FZR120	005	3357	1757	1748
FZR130	005	335F	1771	
FZSBN1	001	34DF	2197	2070 2089 2131
FZSCAJ	001	34E1	2200	2128
FZSCAT	001	34E5	2218	2169
FZSCNT	004	34C7	2246	2040* 2063* 2070* 2096* 2106* 2136* 2146* 2161
FZSDAC	002	35D8	2418	2369* 2373 2375 2375*
FZSDC1	001	35CB	2402	2369



## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 318

SYMBOL	LEN	VALUE	DEFN	REFERENCES
FZSDC5	001	35CC	2403	2279
FZSEXB	004	35D1	2408	2354
FZSLXB	001	0004	2407	2349 2354 2387 2389 2408 2415
FZSLXM	001	0002	2417	2363* 2369 2373 2373 2375 2375 2418
FZSNXZ	001	34E0	2199	2083
FZSPAL	001	0000	2244	2110* 2334* 2340* 2345*
FZSPCH	002	36FB	2703	2486
FZSPDA	002	34E4	2203	2179
FZSPRT	001	3400	2035	
FZSP1B	001	3400	2026	2030
FZSP2B	001	3500	2269	2273
FZSP3B	001	3600	2442	2446 2703 3766
FZSP4B	001	3700	2727	2731
FZSSAJ	001	34E2	2201	2143
FZSXWK	004	35D6	2415	2354* 2363* 2373* 2380 2382 2382* 2389
FZS010	003	3404	2040	
FZS020	004	340E	2047	
FZS030	003	3415	2058	
FZS035	003	3418	2059	
FZS040	003	3424	2063	2060
FZS050	003	3427	2068	
FZS060	003	3434	2075	2069
FZS070	004	3440	2082	
FZS080	003	344B	2087	2084* 2089* 2090 2091 2096
FZS090	003	3455	2090	2082* 2083*
FZS100	004	345B	2096	
FZS110	004	3462	2103	2076 2078 2088
FZS120	003	346D	2110	2058* 2061* 2071 2097
FZS130	003	3473	2122	2048
FZS140	005	3479	2128	
FZS150	004	3481	2131	2134
FZS155	003	3488	2133	2130* 2131* 2136
FZS158	004	348E	2136	2132
FZS160	005	3495	2143	2123
FZS170	003	34A1	2150	2138
FZS180	003	34A4	2160	2043 2112
FZS190	004	34B1	2164	2160* 2161*
FZS2BX	001	35D2	2414	2356* 2357* 2364* 2365* 2371
FZS2B1	001	35CA	2401	2282 2306 2313 2335 2342 2383
FZS2XZ	001	35CD	2405	2307 2357
FZS200	004	34B5	2168	2162
FZS210	005	34C1	2171	2170*
FZS230	004	34C6	2175	2246
FZS240	004	34D5	2184	
FZS260	004	34DB	2189	
FZS3B2	001	36F0	2675	2518
FZS3CC	003	36DB	2683	2467* 2497* 2510* 2512 2514* 2515 2518* 2526* 2527 2533* 2534 2572* 2574 2595* 2605* 2649* 3767* 3768
FZS3CR	001	36F6	2697	2586 2655
FZS3PA	001	36F5	2694	2462*
FZS3PC	001	36F3	2693	2461* 2467 2484 2512 2534* 2545 2559* 2568* 2572 2616*
FZS3PF	001	36F2	2692	2460* 2550* 3774* 3778*
FZS3PL	001	36F2	2686	2620 2692 2693 2694 3775
FZS3PZ	001	36F1	2677	2514
FZS3RM	003	366A	2682	2456* 2574 2650 3768
FZS300	001	3500	2278	2104



## CROSS REFERENCE

SYMBOL	LEN	VALUE	DEFN	REFERENCES	VER 15, MOD 00	31/05/21	PAGE 319
FZS310	003	350E	2286	2280			
FZS320	003	351A	2297				
FZS330	004	3524	2300	2297* 2298*			
FZS340	004	3528	2305	2299			
FZS350	003	3534	2308	2305* 2306* 2307*			
FZS360	003	3537	2312				
FZS370	004	353A	2313	2315			
FZS380	003	353E	2314	2312* 2313* 2319			
FZS390	005	3544	2319				
FZS4B2	001	37DE	2953	2801			
FZS4CC	003	37C6	2962	2751* 2780* 2793* 2795 2797* 2798 2801* 2809* 2810 2816* 2817 2855*			
				2857 2878* 2888* 2923*			
FZS4CR	001	37E4	2976	2869 2929			
FZS4PA	001	37E3	2973	2746*			
FZS4PC	001	37E1	2972	2745* 2751 2768 2795 2817* 2828 2842* 2851* 2855 2899*			
FZS4PF	001	37E0	2971	2744* 2833*			
FZS4PL	001	37E0	2965	2903 2971 2972 2973			
FZS4PZ	001	37DF	2955	2797			
FZS4RM	003	3764	2961	2740* 2857 2924			
FZS400	004	354D	2333	2287 2289			
FZS410	003	3558	2340				
FZS420	004	355E	2342	2344			
FZS430	003	3562	2343	2341* 2342* 2350 2388			
FZS435	003	3568	2345				
FZS440	004	356B	2349				
FZS450	004	3574	2354				
FZS460	003	3586	2363				
FZS470	004	3590	2369	2359			
FZS472	003	3597	2371	2370* 2374 2374* 2376			
FZS474	004	35A1	2374	2372			
FZS480	003	35AC	2380	2358			
FZS490	003	35BB	2387	2381			
FZS500	004	35C2	2389	2387* 2388*			
FZS510	004	35C6	2393				
FZS600	006	3600	2451	2185			
FZS605	003	362B	2474	2473*			
FZS610	003	362E	2484	2220 2220 2221 2222 2223 2225 2226 2227 2228 2230 2230 2231			
				2232 2232 2233 2235 2235 2236 2237 2237 2238			
FZS615	004	3642	2489	2488*			
FZS620	003	3646	2497	2221			
FZS630	003	364C	2510	2222			
FZS632	004	364F	2512	2516			
FZS633	003	365A	2515	2703			
FZS634	004	3660	2518	2513			
FZS636	005	3664	2526	2498			
FZS638	003	3669	2527	2682			
FZS640	005	366F	2533				
FZS650	003	367B	2545	2223 2233 2238			
FZS655	003	3681	2550	2528			
FZS660	003	3687	2559	2225			
FZS670	003	368D	2568	2226			
FZS675	004	3690	2572	2560			
FZS680	003	36A1	2586	2227 2546			
FZS690	003	36A7	2595	2228			
FZS695	003	36B0	2605	2231			
FZS700	003	36B6	2616	2236			

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 320

SYMBOL	LEN	VALUE	DEFN	REFERENCES
FZS710	003	36B9	2620	2535 2551 2575 3770
FZS720	003	36BC	2621	2587
FZS730	006	36BF	2625	2597
FZS740	004	36C8	2630	2452
FZS750	004	36CE	2635	2485 2576 2626
FZS760	003	36D2	2645	2469 2573 2596 2606
FZS770	003	36DA	2650	2683
FZS780	003	36E3	2662	2621
FZS790	004	36EC	2668	2645* 2651 2662*
FZS800	005	3700	2736	2631 3782
FZS805	003	3725	2758	2757*
FZS810	003	3728	2768	
FZS820	003	3740	2780	
FZS830	003	3746	2793	
FZS832	004	3749	2795	2799
FZS834	004	375A	2801	2796
FZS836	005	375E	2809	2781
FZS838	003	3763	2810	2961
FZS840	005	3769	2816	
FZS850	003	3775	2828	
FZS855	003	377B	2833	2811
FZS860	003	3781	2842	
FZS870	003	3787	2851	
FZS875	004	378A	2855	2843
FZS880	003	379B	2869	2829
FZS890	003	37A1	2878	
FZS895	003	37AA	2888	
FZS900	003	37B0	2899	
FZS910	003	37B3	2903	2769 2818 2834 2858
FZS920	003	37B6	2905	2870
FZS950	004	37B9	2909	2770 2859 2880
FZS960	003	37BD	2919	2753 2856 2879 2889
FZS970	003	37C5	2924	2962
FZS980	003	37CE	2937	2905
FZS982	004	37D4	2940	2736*
FZS984	002	37D9	2941	2939*
FZS990	004	37DA	2945	2919* 2925 2937*
FZS991	005	5359	3767	
FZS992	004	5368	3772	3769
FZS993	003	5372	3775	3773
FZS994	004	538D	3783	3780
FZXBCA	001	0DC8	1506	9980* 9985 9998 0044
FZXBKT	001	31CF	1233	1237 1238 1239 1241
FZXBLK	001	31E2	1235	1128
FZXBLN	002	2CE5	0293	0143* 0147 0152 0152*
FZXBPT	001	00FF	1508	0501* 0502* 0503 1081 1202*
FZXBVA	002	2B91	0066	9978 0055
FZXB10	001	32E6	1482	1397
FZXCNT	001	0D56	0660	0496* 0509*
FZXCNV	001	3100	1078	
FZXCRP	001	2CFB	0322	0224
FZXCRR	001	31E1	1239	1128* 1143 1160
FZXCR1	001	31D0	1238	1129
FZXDAC	004	2CE9	0294	0145* 0149 0154 0154*
FZXD1N	004	2CED	0297	0144* 0149*
FZXDTC	001	2EF9	0652	0494* 0595* 0602* 0606 0609*

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 321

SYMBOL	LEN	VALUE	DEFN	REFERENCES
FZXDTM	001	0080	0653	0606 0609
FZXDXL	001	0002	1035	0948 1036 1489
FZXECA	004	2C68	0279	0228
FZXELN	001	2CEA	0296	0182
FZXER0	001	00F0	1510	0533 0605 0708
FZXER1	001	00F1	1511	0608 0744
FZXER2	001	00F2	1512	0546 0614
FZXER3	001	00F3	1513	0486
FZXER4	001	00F4	1514	0474
FZXETS	001	0D58	0655	0455* 0488* 0698 0699
FZXEVA	002	2CE0	0278	0163 0201
FZXEXP	001	32EA	1495	1307* 1333* 1348* 1404* 1406* 1413*
FZXGCS	001	2E00	0451	
FZXICA	001	0003	1251	1188
FZXICB	001	31D0	1241	1093* 1098* 1175* 1182
FZXICC	001	2FB6	0834	0777*
FZXICL	001	0004	0820	0771 0776 0784 0786 0788 0823 0826 0835
FZXICN	001	0001	1250	1182
FZXICR	004	2FB9	0836	0778
FZXICT	001	31E3	1253	1180
FZXICW	004	2FB9	0835	0771 0771* 0776 0776* 0784 0786 0788 0836
FZXIEX	003	2FAD	0823	0784
FZXIPI	002	2FB1	0826	0786
FZXIP1	001	2B00	9974	
FZXIP2	001	2B6E	0040	
FZXIS2	004	2FB5	0828	0788
FZXITL	001	0004	1249	1180 1181
FZXLVA	001	2B8F	0065	0008
FZXMIS	001	2E17	0469	
FZXMNR	001	32F1	1497	1308* 1350 1420
FZXMN1	001	32EB	1496	1309 1411
FZXPDA	002	2CDE	0276	0237
FZXPEM	001	2C18	0139	
FZXPNP	001	2CF7	0315	0173
FZXPQ1	001	2C00	0109	
FZXPQ2	001	2C06	0118	
FZXPRP	001	2CEF	0301	0122* 0128
FZXPSA	002	2B56	0014	0011
FZXPSP	001	2CF3	0308	0183* 0184
FZXP1B	001	2B00	9961	9965
FZXP2B	001	2C00	0094	0098
FZXP3B	001	2D00	0339	0101 0278 0343
FZXP4B	001	2E00	0438	0442
FZXP5B	001	2F00	0689	0693
FZXP6B	001	3000	0858	0862
FZXP7B	001	3100	1065	1069 1129 1143
FZXP8B	001	3200	1298	1302 1309 1350
FZXQML	001	0002	0285	0303
FZXQM1	001	2CE2	0283	0111
FZXQM2	001	2CE1	0281	0120
FZXSEC	001	00FF	0658	0456 0793 0988
FZXSER	001	0D59	0657	0495* 0516 0793* 0988*
FZXSGN	001	32F1	1498	1369* 1414*
FZXSTC	001	2EF8	0650	0493* 0611*
FZXSTP	005	2EC0	0649	0492* 0603*
FZXSTS	001	31CF	1237	1127* 1146*

CROSS REFERENCE														
SYMBOL	LEN	VALUE	DEFN	REFERENCES							VER 15, MOD 00 31/05/21 PAGE 322			
FZXXCL	001	0002	1024	0867	0889	0903	0969	0971	0981	0983	1025	1026	1027	1033
FZXXCT	002	30EA	1033	0867*	0889*	0903*	0969*	0971*	0981	0983				
FZXXHI	002	30E6	1026	0981										
FZXXLO	002	30E8	1027	0983										
FZXXZR	002	30E4	1025	0867										
FZX010	004	2B1E	9991	0031										
FZX020	004	2B57	0019	0006										
FZX030	004	2B8B	0059	0053										
FZX050	003	2C00	0111	9992										
FZX060	003	2C06	0120											
FZX070	003	2C09	0122	0112										
FZX080	005	2C18	0143	0029										
FZX090	003	2C28	0147	0146*	0150	0150*	0153*	0155						
FZX100	004	2C32	0150	0148										
FZX110	004	2C40	0154	0151										
FZX120	004	2C66	0178	0166*	0190	0279								
FZX130	003	2C82	0192	0194										
FZX140	004	2C9D	0206	0131										
FZX150	003	2CA1	0224	0126	0159									
FZX160	004	2CA4	0228	0171	0180	0195								
FZX170	003	2CA8	0233	0129	0174	0185								
FZX180	004	2CBF	0249	0237*	0238*	0242	0254							
FZX190	002	2CC4	0250	0247*										
FZX2D0	001	2CDB	0273	0144										
FZX2D1	001	2CDC	0274	0145	0193									
FZX200	004	2CCD	0259	0243										
FZX210	004	2CD7	0265	0233*	0255									
FZX250	004	2E00	0455											
FZX260	003	2E17	0473											
FZX270	003	2E20	0485	0020										
FZX280	004	2E26	0488	0475										
FZX290	005	2E30	0492	0461										
FZX300	003	2E4F	0507	0524										
FZX310	005	2E55	0509	0531										
FZX320	003	2E6D	0524	0457*	0473*	0485*								
FZX330	003	2E86	0540	0522										
FZX340	003	2E93	0547	0458*	0489*									
FZX350	004	2E96	0550	0474*	0486*									
FZX360	004	2EA4	0559	0534	0547									
FZX370	003	2E9A	0553	0460*	0517	0536	0541	0561*						
FZX375	003	2EA8	0561	0553	0555									
FZX380	003	2EAF	0586	0507	0528	0545								
FZX390	003	2EB2	0587	0459*	0490*									
FZX4B1	002	2EF7	0643											

VER 15, MOD 00 31/05/21 PAGE 323

FZX530	004	2F35	0744	0704											
FZX540	003	2F39	0745	0699*											
FZX550	003	2F3C	0750	0754	0757										
FZX560	004	2F5D	0771	0719											
FZX570	003	2F61	0773	0779											
FZX580	004	2F7B	0784	0775											
FZX590	004	2F90	0793	0709	0745	0752									
FZX6BX	002	30EC	1037	0969	0971										
FZX6B1	002	30E2	1022	0889	0903										
FZX6DX	002	30EC	1036	0948*	0951	0951*	0952*	0957*	0958*	0959	0959*	0960	0960*	0961	0961*
				0962	0962*	0963	0963*	0964*	1037						
FZX600	004	2F94	0795	0734	0761	0785	0787	0789							
FZX610	003	2F98	0804	0717	0760	0773	0780								
FZX620	003	2F9B	0806	0808											
FZX630	004	2FA4	0810	0804*											
FZX650	004	3000	0867	0732											
FZX660	003	300D	0880	0881											
FZX670	003	3016	0886												
FZX675	004	301C	0889	0891											
FZX680	003	3026	0895	0882											
FZX690	003	302F	0902	0868*	0886*	0905									
FZX7B1	003	31C4	1227	1146	1148										
FZX700	003	3036	0904	0897	0902										
FZX710	003	3042	0912	0913											
FZX720	003	3048	0917	0869*	0896	0906	1012*								
FZX730	003	3066	0939	0935											
FZX740	003	3069	0943	0937											
FZX750	003	3084	0957	0950											
FZX780	003	30A1	0968	0932*	0938*										
FZX790	004	30AB	0971	0968											
FZX8BK	001	32EA	1492	1495	1496	1497	1498								
FZX8BX	002	32E9	1490	1404	1406										
FZX8B1	003	32D8	1481	1333	1348	1353	1398								
FZX8DX	002	32E9	1489	1388*	1389	1389*	1390*	1397*	1398*	1490					
FZX8D0	001	32E7	1483	1308											
FZX800	003	30AF	0976	0870*	0887*	0910*	0928	0970							
FZX810	004	30C0	0988	0917	0944	0982									
FZX820	004	30C4	0990	0976	0984										
FZX830	003	30C8	1004	0880	0890	0904	0912	0933	0939	0949	0953				
FZX840	003	30CB	1006	1008											
FZX850	004	30DD	1014	1004*	1011										
FZX860	003	3100	1080	0047											
FZX863	003	3124	1100	1095											
FZX866	003	3127	1101	1097											
FZX870	004	312D	1112	1092*											
FZX873	003	313A	1127	1087											
FZX876	003	3144	1133	1144	1149										
FZX880	003	3156	1143	1135											
FZX883	004	3160	1147	1129*	1143	1148*									
FZX886	003	316B	1153	1138											
FZX890	004	317C	1161	1158*											
FZX893	003	3183	1174	1102											
FZX896	003	3190	1181	1183											
FZX900	004	31A7	1194	1176*											
FZX903	003	31AB	1195	1197											
FZX906	004	31B4	1201	1080*	1117	1162									

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 324

SYMBOL	LEN	VALUE	DEFN	REFERENCES
FZX910	003	31BF	1213	1082 1100 1154 1174 1195
FZX913	003	31C2	1215	1217 1227
FZX916	004	31CB	1219	1213*
FZX920	003	3200	1307	1115
FZX923	003	320D	1321	1322
FZX926	003	3213	1326	
FZX930	004	321F	1333	1335
FZX933	003	3223	1334	1332
FZX936	003	3229	1339	1363
FZX940	003	322C	1340	1328
FZX943	003	3232	1347	1326* 1339* 1356
FZX946	003	3239	1350	1347
FZX950	004	323F	1352	1309* 1350 1353*
FZX953	003	3247	1355	1351
FZX956	003	3259	1367	1341 1361
FZX960	003	3262	1379	1368
FZX963	004	3271	1389	1392
FZX966	004	3282	1397	1399
FZX970	004	3286	1398	1393
FZX973	003	328D	1403	1367 1448* 1453*
FZX976	004	3297	1406	1403
FZX980	003	329B	1411	1380 1405
FZX984	003	32A7	1418	1412
FZX986	004	32B6	1425	1418*
FZX990	003	32BE	1446	1310 1383
FZX992	003	32D3	1466	1321 1334 1355 1362 1382 1391
FZX994	003	32D6	1468	1450 1470 1481
FZX996	003	32DF	1472	1452
FZX998	004	32E2	1473	1446* 1466*
FZZBM1	001	00FF	3316	3196
FZZBN1	001	4CB3	3288	3162 3198 3272
FZZCDT	002	4CB8	3293	3218 3219
FZZCNT	002	4CBC	3301	3218* 3219* 3232 3241 3247
FZZDPL	001	4CBD	3304	3142* 3148* 3176 3210* 3211* 3220* 3222* 3232* 3236 3236* 3237 3237*
				3243* 3249* 3259* 3260* 3264
FZZHCA	002	4CBA	3299	3154* 3163* 3259
FZZIDM	001	0080	3322	3241
FZZITM	001	0040	3323	3247
FZZLOK	001	0001	3319	3190
FZZLRT	001	0000	3318	3182 3184* 3190
FZZMDY	001	0002	3320	3182 3184
FZZNST	001	4CB6	3292	3220 3222
FZZPGB	001	4C00	3129	3133
FZZSDM	001	0001	3324	3243
FZZSTM	001	0080	3325	3249
FZZSXA	002	4CB5	3290	3154
FZZVPL	001	4C06	3147	
FZZVPS	001	4C00	3141	
FZZ005	003	4C09	3153	3143
FZZ010	004	4C25	3171	3159 3273
FZZ020	003	4C29	3172	3153* 3161* 3162* 3199 3260 3272*
FZZ025	003	4C3E	3190	3177
FZZ030	003	4C44	3196	3185
FZZ035	004	4C4B	3198	3200
FZZ040	004	4C4F	3199	3196* 3198* 3211
FZZ050	004	4C61	3219	3221



## CROSS REFERENCE

SYMBOL   LEN   VALUE   DEFN   REFERENCES   VER 15, MOD 00   31/05/21   PAGE 325

FZZ060	003	4C85	3247	3242													
FZZ070	004	4C8E	3259	3248													
FZZ080	002	4CA1	3267	3265*													
FZZ090	004	4CA2	3272	3183	3191												
I\$ADJX	001	0D56	3718														
I\$ADST	001	0C9D	3673														
I\$BASE	001	0C60	3675	0012													
I\$BRCN	001	117B	3727														
I\$BSET	001	119D	3726	0013													
I\$B1SW	001	0040	3783														
I\$B2SW	001	0020	3785														
I\$CADR	001	144C	3764	7311*	7312	7363	7399	7571	7703	7747	7768*	7796	7809	7810	7848		
				7971	8142	9318	9512	9980	0165	2488	3438	3475	3513	3591	3592		
I\$CALL	001	12B1	3758	4153	4161	4206	4501	4512	4797	4814	5214	5231	5639	5664	5696		
				6211	6215	6220	6224	6370	6382	6394	6839	6882	6904	6921	6943		
				6965	7004	7021	7441	7606	7620	7662	7766	7972	8179	8246	8251		
				8263	8306	8368	8554	8566	8667	8735	9173	9991	9999	0019	0028		
				0046	0260	0513	0731	1114	2103	2184	2630	2663	3776	3781			
I\$CBM1	001	0D43	3694														
I\$CBN1	001	0D3E	3690														
I\$CBN2	001	0D3F	3691														
I\$CBN3	001	0D40	3692														
I\$CBN4	001	0D41	3693														
I\$CFBS	001	0AE3	3741														
I\$CLFA	001	0D4A	3700														
I\$CLVA	001	0D49	3699	6844	7248												
I\$CL1C	001	0D46	3697	6821	7368												
I\$CL1F	001	0D44	3695	6809	7302												
I\$CL2C	001	0D47	3698														
I\$CL2F	001	0D45	3696														
I\$CPG1	001	1600	3655														
I\$CPUF	001	0A27	3737	7604	7901												
I\$CSCT	001	0D5A	3713														
I\$CSSW	001	0010	3787														
I\$CSXA	001	2000	3654	0276	2203	3290											
I\$CUPF	001	0A85	3739	6812	7304	1421											
I\$CVAD	001	1358	3752	7398	7795	9317	2487	3474	3512								
I\$DATA	001	0D53	3681	6988*	1687	1699	1734*	1757*	1771*	1772*							
I\$DAT1	001	0D55	3682	6988													
I\$DMSW	001	0BC1	3735														
I\$ECSW	001	0004	3791														
I\$ERRC	001	0CBC	3680	4137*	4806*	5124*	5160*	5179	5225	5233	5235*	5364*	5624*	6208*	6857		
				7258*	7268*	7292*	7381*	7385*	7403*	7579*	7743*	8150*	8258*	8406*	0025		
				0193*	0197*	0318	0456*	0533	0546	0550*	0554*	0559*	0618*	0708	0744		
				1692*	1752*												
I\$FACT	001	0DD1	3720														
I\$FADD	001	075D	3743	5692	5760	6360											
I\$FATE	001	0DE6	3721	7336													
I\$FATP	001	0DE8	3722	7248*	7249	7263	7273										
I\$FDVD	001	0919	3748	6218	6227	6366											
I\$FMPY	001	082A	3746	4214	5178	5224	5630	5670	5743	5754	5765						
I\$FSUB	001	0751	3744	6364													
I\$FWRK	001	0607	3664	4143*	4194	4196*	4197	4251*	4252	4252*	4253	4258	4279	4288*	4296*		
				4310*	4311	4311*	4321	4343	4343*	4664*	4666	4865*	4866*	4867	4867*		
				4868*	4894*	4901*	4908*	4926*	4927	4935*	4937	4965	4965*	5174*	5177		
				5899*	5909	5918*	5921	5927	5930	6213*	6217	6222*	6226	6359*	6361*		



CROSS REFERENCE															
SYMBOL	LEN	VALUE	DEFN	REFERENCES								VER 15, MOD 00    31/05/21    PAGE 326			
I\$IMCI	001	0DCE	3711	6362	6365	6377*	6407	6408	6445	6458	6474*	6475			
I\$IMLN	001	0DC6	3707												
I\$IMPT	001	0DCC	3710												
I\$INDR	001	0DC5	3706												
I\$INIT	001	0607	3663												
I\$INTR	001	0C5C	3667												
I\$IRSW	001	0CDE	3687												
I\$I700	001	0E24	3749	9981											
I\$LBFR	001	12B6	3759	3511*											
I\$LDBR	001	1329	3756	8919											
I\$LDXR	001	1330	3757	4877	7960	7985	8166	8321	8374	8509	8643	8941	9412	9513	9531
I\$LOCK	001	1354	3754	1701 7362 3738	3539 7569 3751	3749 8141	8172	8666	8921	8943	9511	9979	0164	3514	3590
I\$MDFY	001	1349	3753	7570	7696	7843	7988	8140	8171	8709	3437				
I\$MOD4	001	130B	3750	0011*											
I\$NCPG	001	000A	3775												
I\$NDSW	001	0002	3793												
I\$NISW	001	0080	3781												
I\$NPAG	001	0C68	3668												
I\$PARM	001	0D57	3683	6770* 2319*	7599* 2349*	7605* 2350*	0493 2383*	0660 2461	2042 2473	2106 2745	2128* 2757	2143*	2170	2171*	2175*
I\$PGDS	001	144A	3762												
I\$PGNO	001	1449	3761												
I\$PGTB	001	14CA	3765	3197											
I\$PLRT	001	15E2	3766	3171											
I\$PSTK	001	15CA	3767	0014											
I\$PUB1	001	0DC8	3708	1506											
I\$PUB2	001	0DCA	3709												
I\$RESW	001	0CE9	3688	0010*											
I\$RNMK	001	0001	3703												
I\$RNSW	001	0D5C	3702												
I\$RTRN	001	12D3	3760	4219 6219 7849	4317 6228 8013	4514 6400 8270	4669 6459 8412	4943 7030 8548	4988 7259 8574	5242 7269 8713	5448 7293 8745	5698 7322 8954	5766 7447 0026	5934 7614 0059	6064 7744 0206
				0562 2909	0795 3280	0990 3763	1204 3783	1427	1693	1735	1753	2189	2321	2393	2635
I\$SDCT	001	0D59	3715												
I\$SDPT	001	0DD0	3712												
I\$SFCT	001	0D5A	3716												
I\$SFFO	001	0D5D	3724												
I\$SICT	001	0D5B	3717												
I\$SLLC	001	0BA1	3731	6804	2047	2468	2752								
I\$SLNG	001	0BA2	3730	7282*	1727*										
I\$SNSW	001	0001	3795												

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 31/05/21 PAGE 327

I\$STSW	001	0008	3789												
I\$TFSW	001	0D28	3689												
I\$ULNG	001	0C3A	3734	6853*											
I\$UNLK	001	1350	3755	7420	7444	7613	8255	8269	8411	8712	8950	8952	0056	0202	3758
				3760	3762										
I\$USTK	001	0BB0	3733	6858	7305										
I\$VADR	001	144A	3763	6846*	7242*	7253	7274	7301*	7361*	7397*	7419*	7443*	7568*	7612*	7694*
				7695*	7794*	7841*	7842*	8139*	8254*	8268*	8410*	8698*	8708*	8949*	8951*
				9316*	9321*	9322*	9401*	9402*	9510*	9978*	0055*	0163*	0201*	1188*	1726*
				2486*	3436*	3473*	3491*	3509*	3521*	3589*	3757*	3759*	3761*		
I\$WRK1	001	0D59	3684	7321*	7363*	7426	7434*	7654*	7655*	7811*	7813*	7895	8471*	0009*	0655
				0657	2168*	2462	2746								
I\$WRK2	001	0D5B	3685	7436*	7812*	2179*	2180*	2451	2625	2736	3779				
I\$XAD1	001	0C89	3672	7026											
I\$XAD2	001	0C82	3671	6970											
I\$XAD3	001	0C7B	3670	6862											
I\$XAD4	001	0C74	3669												
I\$XERR	001	0CAB	3674												
I\$XIAR	001	0D4C	3679	6759	6769	7241	7314								
I\$XPAG	001	0C61	3678	7313											
I@APRC	001	0006	3857	2077	2279	2279*	2288	2297	2300	2300*	2312	2333	2333	2333*	2341
I@APRL	001	000B	3834												
I@APRS	001	0006	3811	3857											
I@ASTA	001	0000	3869												
I@ASTL	001	0020	3845												
I@ASTS	001	0000	3822	3869											
I@CMEQ	001	0004	3926												
I@CMHI	001	0008	3927												
I@CMLO	001	0002	3925												
I@DEXP	001	0000	3904	4486	4499	4500*	4511*	4655*	4668*	4804	4848*	4938*	4939*	4985	5167
				5374	5391	5392*	5396*	5397*	5398*	5595	5598*	5610	5622	5634	6195
				6206	6342	6350	2075	2077	2082	2282*	2286	2288	2298	2305	2335*
				2356	2365										
I@ICBA	001	F500	3871	1257	1261	1265	1269	1273	1277						
I@ICBL	001	F000	3847												
I@ICBS	001	F500	3824	3871											
I@IVBA	001	F531	3872												
I@IVBL	001	F049	3848												
I@IVBS	001	F531	3825	3872											
I@LCRF	001	0012	3886	3887	7640	7646	7646	7646*	1128	1239	2130				
I@LCRV	001	0013	3887	6804	6853	7728	8331	1160	1160*	1234	1727	2047	2468	2752	
I@LFPZ	001	0012	3956	2160	2163*	2164	2164*	2497	2515	2559	2595	2605	2616	2780	2798
				2842	2878	2888	2899								
I@LPFL	001	0009	3836	3839	3840	7721	7788	8417	8421						
I@LPFS	001	0005	3813	3816	3817	3859	7714	7788	8311	8417	8421				
I@LPFV	001	0005	3859	7334	1257	1261	1265	1269	1273	1277					
I@LPPZ	001	0003	3955	2510	2568	2677	2793	2851	2955						
I@LPSW	001	0080	3844												
I@LSFV	001	0008	3861	5411*	5412*	5413*	5422	5424	5439	5440	5440	5457			
I@LUFL	001	0010	3837	3841											
I@LUFS	001	0008	3814	3818	3860										
I@LUFV	001	0008	3860	3861	3896	3898	3899	3901	3902	4107	4118	4143*	4143*	4173	4176
				4176	4181	4181	4194	4194	4196	4196*	4197	4197	4213	4229	4237
				4251*	4252	4252	4252*	4253	4258	4258	4258	4279	4279	4279	4279
				4288	4288	4288*	4296	4296	4296*	4296*	4310	4310*	4311	4321	4321
				4321	4321	4328	4343	4343	4343	4343	4343*	4343*	4347	4359	4360

## CROSS REFERENCE

SYMBOL   LEN   VALUE   DEFN   REFERENCES   VER 15, MOD 00   31/05/21   PAGE 328

				4361	4362	4363	4364	4365	4366	4367	4652	4833	4835	4835	4855
				4855	4865	4865*	4865*	4894	4894	4894*	4894*	4901	4901	4908	4908
				4908*	4908*	4910	4926	4965	4965	5174	5177	5210	5220	5252	5408
				5408	5408	5410	5422	5424	5439	5441	5461	5462	5629	5669	5685
				5741	5742	5744	5745	5752	5753	5759	5761	5764	5770	5771	5773
				5899	5899*	5901	5927	5930	6063	6198	6200	6210	6213	6213*	6213*
				6214	6217	6217	6217	6222	6222*	6222*	6223	6226	6226	6226	6358
				6359	6359*	6359*	6361	6361*	6361*	6361*	6362	6362	6362	6363	6365
				6365	6365	6365	6384	6384	6389	6389	6389	6390	6390	6405	6406
				6407	6407	6408	6408	6408	6414	6437	6448	6458	6488	6488	6490
				6497	6498	6499	6500	1420	1420*	1493					
I@LXPT	001	0060	3947												
I@MANL	001	0001	3905	4133	4172*	4176*	4642	4987*	5112	5117	5129	5132*	5366	6191	2068
				2281*											
I@MANR	001	0007	3906	5122	5143	5149	5158	5189	5191*	5241*					
I@NCPG	001	000A	3949	3153											
I@NERR	001	0000	3958	5179	5225	5233	6857	0025	0197	0495	0516	0559			
I@NXPG	001	0020	3946	3947											
I@NXPT	001	0003	3945	3947											
I@PEXL	001	0008	3840	7718	7724*	7725	7725*	8392*	8393	8393*	8401				
I@PEXP	001	0004	3864												
I@PEXS	001	0004	3817	3864	7718*	7724	8392	8401*							
I@PMNR	001	0003	3863												
I@PMN1	001	0000	3883												
I@PMRL	001	0007	3839												
I@PMRS	001	0003	3816	3863											
I@PRCL	001	000F	3833	3837											
I@PRCS	001	0007	3810	3814	3856										
I@PREC	001	0007	3856	4143	4194	4234	4236	4253	4254*	4258	4305	4311*	4332	4333	4348
				4490	4629	4637	4664	4664*	4666	4666	4827	4846	4849	4849	4866*
				4867	4867	4867*	4901*	4906	4926*	4937	4937	4950	4965	4965*	5131
				5131	5152	5174*	5177	5201	5379	5386	5409	5441	5457	5468	5595
				5597	5597	5610	5718	5722	5777	5783	5785	5788	5790	5793	5901
				5909	5927	6042	6416	6491	7902	7902	7902*	7907*	7917*	7925*	7933
				7936*	7940	1308	2084								
I@PRSW	001	0087	3868	7717	7722	8387	8396								
I@PRTE	001	0000	3943												
I@RSE1	001	0007	3896	4135	4143	4174	4177*	4179	4186*	4188	4489*	4490	4490*	4503*	4664
				4666*	4667*	4849	4849*	4937*	4983	5131	5131*	5174	5379	5386	5409*
				5597	5597*	5614	5617*	5647*	5651	5653*	5654	5656*	5658	5663*	5691*
				5695*	5741	5742	5744	5752*	5761	5899	6042*	6052	6057	6057*	6063*
				6198	6210	6213	6214*	6222	6223*	6359	6361	6362*	6396*		
I@RSE2	001	000F	3899	4197*	4198*	4213*	4627	4628*	4820	4827	4833	4835*	4846	4855*	4865
				5169	5177*	5210	5220*	5379*	5380*	5386*	5422*	5439*	5629*	5669*	5685*
				5742*	5753*	5759*	5764*	5900*	5901	5901*	5927*	5928*	5930*	6217*	6226*
				6358*	6363*	6365*	6384*	6389*	6390*	6438*	6471*	6472	6473*	6488	
I@RSE3	001	0017	3902	4636*	4637	4637*	5410*								
I@SCOD	001	0000	3936												
I@SGNL	001	000F	3842												
I@SGNS	001	0007	3819	3866											
I@SIDX	001	0001	3937												
I@SIGN	001	0007	3866	5362	6344	6345*	1498	2059	2062*						
I@SPSW	001	0087	3821	3868											
I@STAT	001	0000	3880	6851	7602	7634	7639*	7710	7719*	7723*	7896	7903*	7908*	7915	7916*
				7923	7932*	7935*	8383	8391*	8400*	8665*	1237	1238	2122	2145*	2146
I@SVAD	001	0001	3935	6846	7287	7299*	7300	7301							

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 329

SYMBOL	LEN	VALUE	DEFN	REFERENCES
I@UEXP	001	0000	3882	3904 1495
I@UMNR	001	0007	3865	3906 1497
I@UMN1	001	0001	3884	3905 1496
I@UMRL	001	000F	3841	3842
I@UMRS	001	0007	3818	3819 3865
I@XBRC	001	0003	3919	
I@XCNT	001	0001	3917	6900
I@XCOD	001	0001	3918	6770 6938 6960
I@XLNO	001	0002	3915	
I@XOPC	001	0000	3914	3915 3916 3917 3918 3919 6763 1707
I@XVAD	001	0002	3916	6837 7242 7313* 7314* 7315* 1726 1747 1757
I@1SE1	001	0000	3895	3896 3898 4133 4486 4498* 4499 4499* 4500 4500* 4511* 4642 4655* 4668* 4804 4848* 4938* 4939* 4985 4987* 5112 5132* 5158 5189 5191* 5241* 5362 5366 5374 5385* 5391 5391* 5392* 5393 5393* 5394 5394* 5395 5395* 5396 5396* 5397 5397* 5398* 5425* 5595 5598* 5610 5622 5634 6050 6052* 6063 6191 6195 6206 6342 6344 6345* 6350 6376
I@1SE2	001	0008	3898	3899 3901 5117 5122 5129 5143 5149 5151 5167 5387* 5690*
I@1SE3	001	0010	3901	3902 4172* 4173 4173* 4176* 4181* 4188* 4194* 4196 4198
IBR810	003	1ACF	7009	
IDFADF	001	1A95	6917	7042
IDFBAT	001	1AE0	7038	6764
IDFCLS	001	1AD2	7017	7045
IDFGET	001	1A40	6833	7039
IDFILE	001	1A00	6758	7039 7040 7041 7042 7043 7044 7045 7046 7047
IDFINI	001	1A87	6895	7041
IDFPRS	001	1A9E	6934	7046
IDFPRU	001	1AAD	6956	7047
IDFPUT	001	1A75	6871	7040
IDFRSR	001	1AC0	6984	7043
IDFRST	001	1AC9	7000	7044
IDFSMK	001	000C	6976	6960
IDF010	004	1A04	6763	
IDF020	004	1A0B	6765	6763*
IDF030	004	1A0F	6769	
IDF040	003	1A18	6774	6765*
IDF050	003	1A1B	6800	6877 6939 6961
IDF055	004	1A1E	6804	
IDF060	006	1A25	6809	
IDF065	004	1A2F	6812	6813 6815 6876* 6878*
IDF070	006	1A36	6821	6805
IDF075	004	1A3C	6825	6800* 6817
IDF100	004	1A40	6837	
IDF110	002	1A49	6840	6837*
IDF120	006	1A4A	6844	
IDF130	003	1A59	6850	
IDF140	004	1A66	6857	6852
IDF150	004	1A6E	6862	
IDF200	003	1A75	6876	
IDF220	003	1A84	6887	
IDF300	005	1A87	6900	
IDF310	004	1A8C	6904	
IDF320	003	1A92	6909	
IDF420	004	1A95	6921	
IDF430	003	1A9B	6926	
IDF500	003	1A9E	6938	
IDF510	004	1AA4	6943	

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 330

SYMBOL	LEN	VALUE	DEFN	REFERENCES
IDF520	003	1AAA	6948	
IDF600	003	1AAD	6960	
IDF610	004	1AB3	6965	
IDF620	004	1AB9	6970	6887 6909 6926 6948
IDF700	006	1AC0	6988	
IDF710	003	1AC6	6992	
IDF800	004	1AC9	7004	
IDF900	004	1AD2	7021	
IDF910	004	1AD8	7026	6992 7009
IDF990	004	1ADC	7030	6863 6971
IDIBM2	002	1BA3	7330	7250
IDIFNC	001	1B00	7240	
IDIFTE	002	1BA8	7336	7263
IDIFVA	001	0001	7348	7251 7253 7274*
IDILBI	001	1BA4	7332	7299 7311
IDILFI	001	1BA5	7333	7315
IDILPV	001	1BA6	7334	7320
IDIVAD	002	1BAA	7342	7300* 7320* 7321
IDI010	006	1B09	7248	
IDI020	003	1B13	7250	7254
IDI030	004	1B24	7258	
IDI040	005	1B2C	7263	7252
IDI050	004	1B34	7268	
IDI060	004	1B3C	7273	7264
IDI070	004	1B45	7281	
IDI080	003	1B51	7287	
IDI090	004	1B57	7292	
IDI100	004	1B5F	7299	7288
IDI110	005	1B7E	7311	
IDI130	004	1B95	7320	
IDP210	004	1A7E	6882	
LPBUFR	001	4F00	3710	3696* 3709 3717
LPRCMD	005	4DE9	3557	3627
RETURN	001	4DB3	3527	9369 9534
SFACTR	001	1CF6	7459	7367* 7373* 7379 7383 7390
SFADFR	001	1C00	7355	7356
SFAD2D	001	1CF4	7457	7407* 7408 7427
SFAVD1	002	1CEE	7452	7360 7419
SFAVD2	002	1CF0	7453	7361
SFAWK1	002	1CF8	7460	7360* 7397 7400* 7443
SFA0B0	001	00B0	7451	7408 7445
SFA001	001	1CF1	7454	7369 7373 7374
SFA007	001	1CF2	7455	7391
SFA008	001	1CF3	7456	7393
SFA010	004	1C21	7370	7375
SFA020	003	1C37	7379	7372 7446*
SFA030	003	1C44	7383	7380
SFA032	001	1CF5	7458	7407
SFA040	005	1C51	7390	7384
SFA050	005	1C65	7397	7421
SFA060	003	1C76	7401	7411
SFA065	004	1C7C	7403	7414
SFA070	005	1C83	7405	7392* 7402
SFA075	003	1C8F	7408	7418* 7445*
SFA080	004	1C9B	7412	7409
SFA090	003	1CA5	7418	7413

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 331

SYMBOL	LEN	VALUE	DEFN	REFERENCES
SFA100	003	1CB4	7425	7406
SFA110	004	1CCC	7435	7425*
SFA115	005	1CDA	7443	7404 7430
SFA120	003	1CE3	7445	7382 7386
SFGBLK	003	003D	8281	8217 8224
SFGBS1	001	2100	8136	8137 8247 8252
SFGBS2	001	2200	8288	8289
SFGBS3	001	2300	8436	8437 8588 8600
SFGBVA	002	214B	8167	8165* 8168 8254
SFGCBA	002	21FC	8276	8170* 8240
SFGCBP	001	00FF	8279	8173
SFGCBV	002	2368	8510	8507* 8508*
SFGCNL	002	22E8	8424	8311* 8316* 8331* 8333 8334 8359* 8425
SFGDEH	001	0006	8611	8521
SFGDLS	001	22E3	8417	8316
SFGDRL	001	00E9	8600	8470
SFGDWL	001	00E3	8588	8462
SFGD2P	004	2276	8356	8318*
SFGELS	001	0004	8421	8393
SFGETR	001	2100	8138	
SFGHDL	001	0007	8610	8521 8611 8614
SFGICR	003	0040	8280	8183 8235
SFGLEH	001	23F4	8613	8523 8530* 8535 8538 8543* 8544
SFGMFA	006	2272	8350	8344*
SFGMLQ	002	22EC	8428	8338* 8339* 8340 8342 8346 8429
SFGMS1	001	00FF	8420	8338
SFGMTA	006	2270	8349	8328* 8346* 8376*
SFGNFM	001	00FF	8278	8161 8164
SFGONE	001	22E4	8418	8376
SFGPAF	001	23F1	8605	8516
SFGPCL	002	22EA	8427	8333* 8336* 8339 8357 8358 8359
SFGPLR	001	23E9	8593	8600
SFGPLW	001	23E3	8581	8588
SFGPSL	001	23F3	8607	8542 8543
SFGRPL	004	2334	8479	8469* 8470*
SFGRST	003	003A	8282	8218
SFGSA0	001	0F00	8609	
SFGSBR	004	233A	8485	8472*
SFGSB2	007	23FA	8615	8552* 8557 8563* 8564
SFGSDF	002	22E6	8423	8317* 8334 8336 8371*
SFGSHD	007	23FA	8614	8521* 8615
SFGSSL	001	23F2	8606	8529 8530
SFGSSZ	002	23F0	8604	8499
SFGSXR	004	233E	8487	8473* 8506* 8515 8522
SFGVCB	002	2234	8322	8319* 8320*
SFGVD2	002	21FA	8274	8139 8268
SFGVNB	002	229D	8375	8372* 8373* 8410
SFGWPL	004	231E	8468	8461* 8462*
SFGXRD	001	00FE	8616	8564*
SFGZRO	002	22E2	8416	8301
SFG120	004	2126	8150	8147
SFG150	003	212D	8152	8149
SFG200	003	2130	8156	8145
SFG205	004	2142	8165	8162
SFG210	003	215A	8174	8163* 8175*
SFG215	004	2160	8179	8242



## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 332

SYMBOL	LEN	VALUE	DEFN	REFERENCES
SFG220	003	216F	8188	8231
SFG225	003	2172	8189	8183* 8217 8218* 8224* 8235*
SFG227	003	2175	8191	8280 8281 8282
SFG230	003	218A	8204	8210
SFG235	003	219F	8217	8202
SFG240	004	21A8	8221	8226*
SFG245	003	21AF	8224	8282
SFG250	003	21B2	8226	8281
SFG255	003	21B5	8230	8189 8213 8219 8222 8236 8280
SFG260	003	21BB	8235	8192
SFG265	003	21C1	8240	8195
SFG270	003	21C4	8242	8184* 8197*
SFG280	004	21CD	8251	8174
SFG282	005	21D3	8254	8259
SFG285	004	21DF	8258	8207
SFG290	004	21E6	8263	8157
SFG295	005	21EC	8268	8151 8256
SFG450	003	220D	8311	8302
SFG470	004	2220	8317	8314
SFG500	004	2249	8333	8330 8377
SFG520	003	2258	8338	8335
SFG550	006	226D	8348	8340* 8349 8350
SFG555	004	2273	8355	8356
SFG570	004	22A5	8382	8360
SFG575	003	22AF	8386	8312* 8315*
SFG585	003	22C3	8396	8386
SFG690	004	22D0	8406	8324
SFG695	005	22D4	8410	8384 8387 8394 8396 8402
SFG750	003	2300	8451	8539
SFG760	004	230C	8455	8517
SFG780	003	2313	8461	8454
SFG785	004	2319	8464	8468
SFG790	004	232F	8475	8479
SFG795	004	2337	8484	8485
SFG800	004	233B	8486	8487
SFG810	003	2345	8496	8456
SFG825	003	2355	8502	8497
SFG830	003	2358	8506	8452
SFG840	004	2379	8521	8513
SFG850	003	2386	8528	8537* 8541*
SFG860	003	2394	8535	8524
SFG870	004	239D	8538	8528
SFG880	003	23A4	8541	8536
SFG890	004	23AF	8544	8531
SFG900	004	23B3	8548	8500
SFG920	003	23B7	8552	8247
SFG930	003	23C6	8562	8252 8569* 8572*
SFG935	004	23D0	8566	8562
SFG940	003	23DC	8572	8558
SFG945	004	23DF	8574	8570
SFPBFR	006	1EC8	7753	7747* 7751*
SFPBS1	001	1D00	7564	7565
SFPBS2	001	1E00	7681	7682
SFPBS3	001	1F00	7792	7793 7861 7871
SFPBS4	001	2000	7881	7882
SFPCBP	002	2094	7962	7888* 7984



## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 333

SYMBOL	LEN	VALUE	DEFN	REFERENCES
SFPCBV	002	2095	7961	7962 7963
SFPCFL	005	20DE	8004	7895* 7940* 7945 8000 8001 8011
SFPCNL	001	1EF2	7773	7714* 7721* 7728* 7735 7746 7757* 7774 7811
SFPCPT	002	20C0	7987	7944 7965 7980* 7989 7996* 8010
SFPCPW	002	20FD	8028	7944* 7945* 7949
SFPCRT	002	1DF0	7677	7592
SFPCXI	004	1DE0	7660	7626*
SFPC01	002	1EFC	7783	7741 7759
SFPDAC	002	20FD	8026	7921* 7925 7927 7927* 8027
SFPDCA	005	20DF	8003	8001*
SFPDEV	002	1DEB	7671	7585* 7608 7673
SFPDIC	002	1DEB	7673	7640* 7642*
SFPDLS	001	0004	7788	7725
SFPDP1	001	1F7E	7854	7861
SFPDP2	001	1F84	7864	7871
SFPD1D	001	007E	7861	7815
SFPD2D	001	0084	7871	7822
SFPENC	001	0005	8019	7940
SFPEXI	004	20FA	8023	7907
SFPEZR	001	20FB	8024	7908
SFPLEX	001	0004	8017	7907 8019 8023
SFPLXM	001	0002	8018	7921 7925 7925 7927 7927
SFPMPT	002	1DEE	7676	7588
SFPMS1	001	00FF	7789	7748
SFPMVL	006	1EC6	7754	7748* 7749* 7750 7751
SFPNGE	002	20FD	8027	7914* 7915* 7916 8028
SFPONE	001	1DEC	7675	7642
SFPPRT	002	1EF8	7779	7734* 7746* 7749 7756 7757 7780
SFPRT2	002	1F8B	7875	7794
SFPSAO	001	0F00	7878	
SFPSCA	002	1EFA	7782	7768 7848*
SFPSIO	002	1EF6	7787	7699* 7712
SFPSTC	005	20E1	8005	7894* 8000*
SFPSTK	006	1ECA	7755	7729* 7750* 7759* 7812
SFPUTR	001	1D00	7567	
SFPVCA	002	20C0	7986	7887* 7984* 7987
SFPVD2	002	1DF2	7678	7568 7612
SFPWK2	002	1EF6	7778	7737* 7739* 7740* 7741* 7787
SFPWRK	001	1EF4	7777	7697* 7698* 7732 7734 7735* 7738* 7739
SFPXR1	004	1E7C	7731	7700* 7705
SFPX01	001	20F5	8021	7996 8012
SFPZD1	001	20F6	8022	7921
SFP050	004	1D26	7579	7576
SFP075	003	1D2D	7581	7578
SFP100	005	1D30	7585	7574
SFP120	003	1D43	7590	7587
SFP130	003	1D54	7597	7589
SFP133	004	1D61	7601	7598
SFP135	004	1D6F	7605	7603
SFP140	004	1D73	7606	7600
SFP150	005	1D7E	7612	7580 7622 7667
SFP175	003	1D8B	7618	7591
SFP200	003	1D9A	7626	7619
SFP220	004	1DA3	7633	
SFP230	003	1DB3	7641	7645 7649
SFP250	003	1DD0	7650	7643

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 334

SYMBOL	LEN	VALUE	DEFN	REFERENCES
SFP320	004	1DDD	7659	7628 7635 7660
SFP350	005	1E0D	7694	7690
SFP370	003	1E1B	7697	7692
SFP385	004	1E3F	7709	7702
SFP400	003	1E5F	7721	7713
SFP410	003	1E65	7723	
SFP430	003	1E73	7728	7711
SFP450	003	1E76	7729	7717 7720 7722 7726
SFP460	004	1E79	7730	7731
SFP480	004	1EA5	7743	
SFP490	004	1EA9	7744	7707 7758
SFP5	001	0005	7877	7814* 7815* 7821* 7822*
SFP500	004	1EAD	7746	7733 7736 7761
SFP510	005	1EB1	7747	7742
SFP550	006	1EC5	7752	7753 7754 7755
SFP560	004	1ED6	7759	
SFP580	003	1EE0	7765	7691 7706 7760
SFP590	004	1EEE	7769	7765*
SFP610	005	1F1E	7809	7798 7800
SFP625	006	1F28	7811	7809*
SFP630	006	1F2E	7812	7810*
SFP635	004	1F3E	7817	7814* 7815*
SFP640	004	1F50	7826	7821* 7822*
SFP650	004	1F58	7834	7823*
SFP655	004	1F5C	7835	7824*
SFP675	005	1F66	7841	7805
SFP680	006	1F74	7848	7796*
SFP720	004	204A	7921	7910
SFP725	003	2051	7923	7922* 7926 7926* 7928
SFP730	004	205B	7926	7924
SFP750	003	2066	7932	7909
SFP760	003	2075	7940	7934
SFP785	004	2078	7944	7897
SFP790	003	208A	7949	7947 7948* 7950 7952
SFP800	004	2090	7960	7891
SFP830	004	20AA	7977	7886* 8009
SFP850	004	20B7	7984	7953
SFP865	004	20D5	8000	7990
SFP875	005	20DD	8002	8003 8004 8005
SFP950	004	20F1	8013	7979
SFRBS1	001	2400	8623	8622 8729
SFRCAL	001	2400	8627	
SFRCLS	001	240A	8634	
SFRIXR	004	2484	8700	8645*
SFRLPR	003	24B7	8733	8744
SFRNOE	001	24AB	8719	8696* 8720
SFRONE	001	24AA	8717	8696
SFRSET	001	240D	8639	
SFRVD2	002	2412	8644	8698 8708
SFRX10	001	24AC	8722	8701
SFR100	004	240D	8643	
SFR110	003	2416	8649	8629* 8710*
SFR115	003	241C	8652	8649 8702
SFR130	003	241F	8656	
SFR135	004	2448	8669	8663*
SFR140	003	244F	8674	8635* 8659 8706*

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 335

SYMBOL	LEN	VALUE	DEFN	REFERENCES
SFR200	004	2452	8679	8688
SFR300	003	2461	8686	8674
SFR900	003	2472	8695	8628* 8657 8661 8682 8711*
SFR950	004	2481	8699	8700
SFR995	003	248C	8706	8695 8697
SFR996	004	24C3	8737	8733*
SFR997	004	24CA	8739	8732
SFR998	006	24D1	8741	8738
SFR999	004	24D7	8742	8740
SF1000	001	24E5	8747	8734
V\$APWR	001	0800	2366	2512
V\$BFR1	001	5400	2429	2620 0066
V\$BFR2	001	5500	2430	2621
V\$CBNZ	001	0CB2	2438	2519 4502 4513
V\$CCON	001	5120	2445	2617
V\$CDCV	001	3100	2442	2572 8567
V\$CDSY	001	2E00	2441	2569 8555
V\$CFPZ	001	0C70	2436	2518 4815 5640 6371
V\$CNXZ	001	0470	2439	2507 4162
V\$CSSR	001	5100	2444	2616
V\$CZFP	001	04AD	2437	2508 4207 5665 6395
V\$DTLN	001	4600	2451	2604
V\$DTVR	001	4700	2452	2605
V\$FABS	001	1761	2337	2536
V\$FACS	001	1400	2353	2528
V\$FASN	001	1413	2352	2529
V\$FATN	001	1100	2351	2525 6383
V\$FCOS	001	0A00	2348	2514 6212 6225
V\$FCOT	001	0D00	2346	2520
V\$FCSC	001	1725	2350	2535
V\$FDEG	001	17DA	2357	2540
V\$FDET	001	4540	2360	2603
V\$FEXP	001	0500	2344	2509 4798 5232
V\$FHCS	001	1500	2356	2530
V\$FHSN	001	1557	2355	2531
V\$FHTN	001	1593	2354	2532
V\$FINT	001	176C	2338	2537
V\$FLGT	001	0200	2342	2502 4154 4878
V\$FLOG	001	0219	2341	2504 5215
V\$FLTW	001	020B	2343	2503
V\$FRAD	001	17CB	2358	2539
V\$FRND	001	1800	2359	2541
V\$FSEC	001	1700	2349	2534
V\$FSGN	001	17A7	2339	2538
V\$FSIN	001	0A1A	2347	2515 6216 6221
V\$FSQR	001	0900	2340	2513
V\$FTAN	001	0D28	2345	2521
V\$IFCI	001	1B00	2329	2545
V\$IFIO	001	1A00	2331	2544
V\$ISDN	001	1900	2330	2542
V\$KBTL	001	1EAC	2473	
V\$KBTS	001	0DAC	2472	
V\$LPRB	001	4F00	2427	2614 9538 3545
V\$LPRT	001	4D00	2425	2612 9324 9514 9532 3541 3750
V\$LPR2	001	4E00	2426	2613 3554 3558
V\$MADD	001	4007	2374	2592

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 336

SYMBOL	LEN	VALUE	DEFN	REFERENCES
V\$MASN	001	43A0	2372	2599
V\$MCON	001	4324	2379	2597
V\$MIDN	001	4300	2380	2596
V\$MINV	001	4500	2384	2602
V\$MMPY	001	4100	2376	2593
V\$MSMY	001	4264	2377	2595
V\$MSUB	001	4000	2375	2591
V\$MTRN	001	4400	2383	2601
V\$MZER	001	432B	2381	2598
V\$PCH1	001	5200	2465	2618
V\$PCH2	001	5300	2466	2619 9445 2703 3562 3563
V\$SCDI	001	2A00	2422	2563 8180
V\$SCDO	001	2A96	2423	2564 7973
V\$SFA2	001	5000	2407	2615 7442
V\$SFD1	001	0000	2417	2500 7452
V\$SFD2	001	0100	2418	2501 7453 7678 8274 8644
V\$SKEY	001	2500	2421	2558 8920 8942 9076 9077 0000
V\$SPRT	001	2800	2420	2561 8736 9174 9413 0261 2664 3777
V\$VMPL	001	4C06	2459	2611
V\$VMPS	001	4C00	2458	2610
V\$XKAF	001	1C00	2406	2546 6922
V\$XKCA	001	2400	2410	2554
V\$XKCL	001	240A	2409	2555 7022
V\$XKIN	001	2B00	2405	2565 6905
V\$XKLP	001	24AD	2411	
V\$XKRS	001	240D	2408	2556 7005
V\$XMGT	001	3E06	2399	2586
V\$XMIN	001	3D00	2398	2584
V\$XMPL	001	3F06	2402	2589
V\$XMPS	001	3F00	2401	2588
V\$XMPT	001	3E0C	2400	2587
V\$XMPU	001	3F13	2403	2590
V\$XMRD	001	3E00	2397	2585
V\$XSGT	001	2100	2392	2551 8247 8252 8264 8307 8369
V\$XSIN	001	2B6E	2391	2566
V\$XSPR	001	3400	2394	2575 6944 7607
V\$XSPT	001	1D00	2393	2547 6883 7621 7663 7767 7875 8668
V\$XSPU	001	3800	2395	2579 6966
V\$XSRD	001	3300	2390	2574
V\$00E1	001	0000	2500	
V\$01E1	001	0100	2501	
V\$02E1	001	0200	2502	
V\$02E2	001	020B	2503	
V\$02F3	001	0219	2504	
V\$03CC	001	0300	2505	
V\$04CC	001	0400	2506	
V\$04E1	001	0470	2507	
V\$04E2	001	04AD	2508	
V\$05E1	001	0500	2509	
V\$06CC	001	0600	2510	
V\$07CC	001	0700	2511	
V\$08E1	001	0800	2512	
V\$09E1	001	0900	2513	
V\$10E1	001	0A00	2514	
V\$10E2	001	0A1A	2515	
V\$11CC	001	0B00	2516	

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 31/05/21 PAGE 337

V\$12CC 001 0C00 2517  
V\$12E1 001 0C70 2518  
V\$12E2 001 0CB2 2519  
V\$13E1 001 0D00 2520  
V\$13E2 001 0D28 2521  
V\$14CC 001 0E00 2522  
V\$15CC 001 0F00 2523  
V\$16CC 001 1000 2524  
V\$17E1 001 1100 2525  
V\$18CC 001 1200 2526  
V\$19CC 001 1300 2527  
V\$20E1 001 1400 2528  
V\$20E2 001 1413 2529  
V\$21E1 001 1500 2530  
V\$21E2 001 1557 2531  
V\$21E3 001 1593 2532  
V\$22CC 001 1600 2533  
V\$23E1 001 1700 2534  
V\$23E2 001 1725 2535  
V\$23E3 001 1761 2536  
V\$23E4 001 176C 2537  
V\$23E5 001 17A7 2538  
V\$23E6 001 17CB 2539  
V\$23E7 001 17DA 2540  
V\$24E1 001 1800 2541  
V\$25E1 001 1900 2542  
V\$26E1 001 1A00 2544  
V\$27E1 001 1B00 2545  
V\$28E1 001 1C00 2546  
V\$29E1 001 1D00 2547  
V\$30CC 001 1E00 2548  
V\$31CC 001 1F00 2549  
V\$32CC 001 2000 2550  
V\$33E1 001 2100 2551  
V\$34CC 001 2200 2552  
V\$35CC 001 2300 2553  
V\$36CC 001 2400 2557  
V\$36E1 001 2400 2554  
V\$36E2 001 240A 2555  
V\$36E3 001 240D 2556  
V\$37E1 001 2500 2558  
V\$38CC 001 2600 2559  
V\$39CC 001 2700 2560  
V\$40E1 001 2800 2561  
V\$41CC 001 2900 2562  
V\$42E1 001 2A00 2563  
V\$42E2 001 2A96 2564  
V\$43E1 001 2B00 2565  
V\$43E2 001 2B6E 2566  
V\$44CC 001 2C00 2567  
V\$45CC 001 2D00 2568  
V\$46E1 001 2E00 2569  
V\$47CC 001 2F00 2570  
V\$48CC 001 3000 2571  
V\$49E1 001 3100 2572  
V\$50CC 001 3200 2573

## CROSS REFERENCE

SYMBOL   LEN   VALUE   DEFN   REFERENCES   VER 15, MOD 00   31/05/21   PAGE 338

V\$51E1   001   3300   2574  
V\$52E1   001   3400   2575  
V\$53CC   001   3500   2576  
V\$54CC   001   3600   2577  
V\$55CC   001   3700   2578  
V\$56E1   001   3800   2579  
V\$57CC   001   3900   2580  
V\$58CC   001   3A00   2581  
V\$59CC   001   3B00   2582  
V\$60CC   001   3C00   2583  
V\$61E1   001   3D00   2584  
V\$62E1   001   3E00   2585  
V\$62E2   001   3E06   2586  
V\$62E3   001   3E0C   2587  
V\$63E1   001   3F00   2588  
V\$63E2   001   3F06   2589  
V\$63E3   001   3F13   2590  
V\$64E1   001   4000   2591  
V\$64E2   001   4007   2592  
V\$65E1   001   4100   2593  
V\$66CC   001   4200   2594  
V\$66E1   001   4264   2595  
V\$67E1   001   4300   2596  
V\$67E2   001   4324   2597  
V\$67E3   001   432B   2598  
V\$67E4   001   43A0   2599  
V\$68E1   001   4400   2601  
V\$69E1   001   4500   2602  
V\$69E2   001   4540   2603  
V\$70E1   001   4600   2604  
V\$71E1   001   4700   2605  
V\$72CC   001   4800   2606  
V\$73CC   001   4900   2607  
V\$74CC   001   4A00   2608  
V\$75CC   001   4B00   2609  
V\$76E1   001   4C00   2610  
V\$76E2   001   4C06   2611  
V\$77CC   001   4D00   2612  
V\$78CC   001   4E00   2613  
V\$79CC   001   4F00   2614  
V\$80E1   001   5000   2615  
V\$81E2   001   5100   2616  
V\$81E3   001   5120   2617  
V\$82E1   001   5200   2618  
V\$83E2   001   5300   2619  
V\$84E1   001   5400   2620  
V\$85E2   001   5500   2621  
V@CDPT   001   0007   2632  
V@CHGH   001   0008   2737  
V@CMIC   001   0002   2633  
V@CMNI   001   00FF   2630  
V@CMUL   001   0007   2738  
V@CNIX   001   0080   2631  
V@COEX   001   001E   2628  
V@CPLS   001   00F0   2635  
V@CPRC   001   000A   2637

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 339

SYMBOL	LEN	VALUE	DEFN	REFERENCES
V@CSQR	001	0003	2735	
V@CSTR	001	0002	2736	
V@CTTA	001	0027	2638	
V@DCAD	001	0002	2658	2659
V@DEXP	001	0000	2663	
V@DMAN	001	000D	2665	2666
V@DMN1	001	0001	2664	
V@DPDF	001	0002	2653	
V@DSAD	001	0001	2654	
V@DSGN	001	000D	2666	
V@DVAD	001	0004	2659	
V@EART	001	0001	2636	
V@ECRT	001	0038	2709	
V@EFUL	001	00F8	2708	
V@EINV	001	00FB	2704	
V@EIPR	001	00F5	2705	
V@ENSV	001	00F7	2706	
V@ENUL	001	0000	2703	
V@ERPC	001	0020	2634	
V@ESAV	001	00F6	2707	
V@FEHN	001	0002	2733	
V@FEPL	001	0091	2729	
V@FERS	001	0003	2732	
V@FPGS	001	0081	2728	
V@FRET	001	0015	2731	
V@FSPC	001	0040	2730	
V@FTAB	001	0000	2734	
V@KADD	001	004E	2719	
V@KCLE	001	006E	2716	
V@KDIV	001	0061	2722	
V@KEMN	001	006C	2714	
V@KEPL	001	006B	2713	
V@KMUL	001	005C	2721	
V@KPER	001	004B	2724	
V@KPST	001	007B	2718	
V@KPWR	001	005A	2723	
V@KSQR	001	006F	2715	
V@KSTO	001	006D	2717	
V@KSUB	001	0060	2720	
V@LAIP	001	0003	2684	2685
V@LDEX	001	0002	2687	
V@LETE	001	0003	2691	
V@LEXP	001	0001	2681	2683
V@LFKO	001	0006	2686	
V@LINI	001	0200	2690	
V@LLKS	001	0010	2683	
V@LMAN	001	000F	2682	2683
V@LNOP	001	0015	2688	
V@LTBE	001	0007	2685	
V@LVPG	001	0100	2689	2690
V@MCHS	001	00C0	2670	
V@MCRD	001	0010	2646	
V@MDEF	001	0008	2647	
V@MEXC	001	0080	2644	
V@MEXT	001	0004	2673	
V@MICC	001	0010	2629	



VER 15, MOD 00 31/05/21 PAGE 340

V@MIPC	001	0080	2671		
V@MIPL	001	0020	2677		
V@MLST	001	0040	2645		
V@MPND	001	0000	2676		
V@MPOF	001	0080	2674		
V@MPRC	001	0020	2643		
V@MSFU	001	0002	2648		
V@MSTN	001	0004	2642		
V@OALL	001	00F4	2699		
V@ONUL	001	00F0	2695	2696	
V@OPM1	001	00F2	2697	2698	
V@ORTN	001	00F1	2696	2697	
V@OSTK	001	00F3	2698	2699	
V@PEOF	001	0002	2672		
V@PSQ2	001	0014	2675		
VLPRT2	001	4E00	3588	3553	3554
VLPRT3	001	5300	3731		
VLPRT4	001	5359	3765		
VLPRT5	001	5391	3785		
VLPRT6	001	53B6	3799		

```
OL105 I THE CODE LENGTH OF #FMSTD IS 21453 DECIMAL.
OL103 I TOTAL NUMBER OF LIBRARY SECTORS REQUIRED IS 58
      NAME-#FMSTD,PACK-R1R1R1,UNIT-R1,RETAIN-P,LIBRARY-R,CATEGORY-000
```

START ADDRESS	CATEGORY	NAME AND ENTRY	CODE LENGTH HEXADECIMAL	DECIMAL
0200	0	#FMSTD	53CD	21453
OL100	I	THE TOTAL CORE USED BY #FMSTD IS 21453 DECIMAL.		
OL101	I	THE START CONTROL ADDRESS OF THIS MODULE IS 0200.		
OL104	I	TOTAL NUMBER OF LIBRARY SECTORS REQUIRED IS 84		
		NAME-#FMSTD,PACK-R1R1R1,UNIT-R1,RETAIN-P,LIBRARY-O		