

ACCESS NAMES TABLE

SOURCE ACCESS NAME= PPC2.P359.SRC.SPEED
OBJECT ACCESS NAME= PPC2.P359.OBJ.SPEEDS
LISTING ACCESS NAME= PPC2.P359.LST.SPEEDS
ERROR ACCESS NAME=
OPTIONS= XREF
MACRO LIBRARY PATHNAME=

LINE	KEY	NAME
0002	A	VERSION =>PPC2.P359.SRC.P359


```
0055          DEF  SPEED
0056          REF  FAC, R1LB, GOTO90, SYNCHK, ERRSYN, CBH63, SET, EXTRAM
0057          REF  PUTSTK, GETSTK, SAVREG, SETREG, PARSE, PGMCHR, RESE
0058          REF  FAC2, FAC10, CFI, CHAT, ERRT, RAMFLG, EXTRAM, STLN
0059          REF  ENLN, GET1, GETG2, RAMTOP
0060 0000
0061          00B3  COMMA$ EQU  >B3          Comma token value
0062          00B7  LPAR$  EQU  >B7
0063 0000
0064 0000 0460  BSYNCH B    @SYNCHK
          0002 0000
0065 0004 0460  BERSYN B    @ERRSYN
          0006 0000
0066 0008 0460  BERSNM B    @ERRT
          000A 0000
```

```
0068 000C      SPEED
0069          *-----CONDITIONAL ASSEMBLY-----*
0070          ASMIF VERS=DX10
0071          MOVB *R15+,RO          Read XML code
0072          ASMELS
0073 000C D01D  MOVB *R13,RO          Read XML code
0074          ASMEND
0075          *-----END OF CONDITIONAL ASSEMBLY-----*
0076 000E 0980  SRL  RO,8           Shift for word usage
0077 0010 13F7  JEQ  BSYNCH          0 is index for SYNCHK
0078 0012 0600  DEC  RO           Not SYNCHK-check further
0079 0014 1344  JEQ  PARCOM          1 is index for PARCOM
0080 0016 0600  DEC  RO           Not PARCOM-check further
0081 0018 1320  JEQ  RANGE          2 is index for RANGE
0082          * All others assumed to be SEETWO
```

```

0084 *****
0085 * Find the line specified by the number in FAC *
0086 * Searches the table from low address(high number *
0087 * to high address(low number). *
0088 *****
0089 001A 020A SEETWO LI R10,SET Assume number will be found
      001C 0000
0090 001E 0207 LI R7,GET1 Assume reading from the VDP
      0020 0000
0091 0022 D020 MOV @RAMTOP,R0 But correct
      0024 0000
0092 0026 1302 JEQ SEETW2 If
0093 0028 0207 LI R7,GETG2 ERAM is present
      002A 0000
0094 002C C0E0 SEETW2 MOV @ENLN,R3 Get point to start from
      002E 0000
0095 0030 0223 AI R3,-3 Get into table
      0032 FFFD
0096 0034 0697 SEETW4 BL *R7 Read the number from table
0097 0036 0241 ANDI R1,>7FFF Throw away possible breakpoint
      0038 7FFF
0098 003A 8801 C R1,@FAC Match the number needed?
      003C 0000
0099 003E 130A JEQ SEETW8 Yes-return with condition set
0100 0040 1B07 JH SEETW6 No-and also passed it=>return
0101 0042 0223 AI R3,-4 No-but still might be there
      0044 FFFC
0102 0046 8803 C R3,@STLN Reached end of table?
      0048 0000
0103 004A 14F4 JHE SEETW4 No-so check further
0104 004C C0E0 MOV @STLN,R3 End of table-default to last
      004E 0048
0105 0050 020A SEETW6 LI R10,RESET Indicate not found
      0052 0000
0106 0054 C803 SEETW8 MOV R3,@EXTRAM Put pointer in for GPL
      0056 0000
0107 0058 045A B *R10 Return with condition
    
```

0109	005A	C30B	RANGE	MOV	R11,R12	Save return address
0110	005C	9820		CB	@FAC2,@CBH63	Have a numeric
	005E	0000				
	0060	0000				
0111	0062	1BD2		JH	BERSNM	Otherwise string-num mismatch
0112	0064	04E0		CLR	@FAC10	Assume no conversion error
	0066	0000				
0113	0068	06A0		BL	@CFI	Convert from float to integer
	006A	0000				
0114	006C	D020		MOVB	@FAC10,R0	Get an error?
	006E	0066				
0115	0070	160E		JNE	RANERR	Yes-indicate it
0116				*-----CONDITIONAL ASSEMBLY-----*		
0117				ASMIF	VERS=DX10	
0118				MOVB	*R15+,R0	Read lower limit
0119				SRL	R0,8	Shift for word compare
0120				MOVB	*R15+,R1	Read 1st byte of upper limit
0121				MOVB	*R15+,@R1LB	Read 2nd byte of upper limit
0122						
0123				ASMELS		
0124	0072					
0125	0072	D01D		MOVB	*R13,R0	Read lower limit
0126	0074	0980		SRL	R0,8	Shift for word compare
0127	0076	D05D		MOVB	*R13,R1	Read 1st byte of upper limit
0128	0078	06C1		SWPB	R1	Kill time
0129	007A	D05D		MOVB	*R13,R1	Read 2nd byte of upper limit
0130	007C	06C1		SWPB	R1	Restore upper limit
0131				ASMEND		
0132				*-----END OF CONDITIONAL ASSEMBLY-----*		
0133	007E	C0A0		MOV	@FAC,R2	Get the value
	0080	003C				
0134	0082	1105		JLT	RANERR	If negative - error
0135	0084	8002		C	R2,R0	Less than low limit?
0136	0086	1103		JLT	RANERR	Yes-error
0137	0088	8042		C	R2,R1	Greater than upper limit?
0138	008A	1B01		JH	RANERR	Yes-error
0139	008C	045C		B	*R12	All OK - so return
0140	008E	06A0	RANERR	BL	@SETREG	Set up registers for error
	0090	0000				
0141	0092	0460		B	@GOTO90	* BAD VALUE
	0094	0000				

```
0143          *      Make sure at a left paren
0144 0096 9820 LPAR  CB   @CHAT,@LBLP$      At a left paren?
      0098 0000
      009A 00AB'
0145 009C 16B3          JNE  BERSYN          No-syntax error
0146 009E
0147          *      Parse up to a comma and insure at a comma
0148 009E 06A0 PARCOM BL   @PUTSTK          Save GROM address
      00A0 0000
0149 00A2 06A0          BL   @SETREG          Set up RB/R9
      00A4 0090'
0150 00A6 06A0          BL   @PARSE          Parse the next item
      00AB 0000
0151 00AA  B3          BYTE COMMA$          Up to a comma
0152 00AB  B7          LBLP$ BYTE LPAR$
0153 00AC-02B8          CI   RB,COMMA$*256    End on a comma?
      00AE B300
0154 00B0 16A9          JNE  BERSYN          No-syntax error
0155 00B2 06A0          BL   @POMCHR          Yes-get char after it
      00B4 0000          MOV B TR14, R9
0156 00B6 06A0          BL   @SAVREG          Save RB/R9 for GPL
      00B8 0000
0157 00BA 06A0          BL   @GETSTK          Restore GROM address
      00BC 0000
0158 00BE 0460          B    @RESET          Return to GPL reset
      00C0 0052'
0159          END
NO ERRORS, 0001 WARNINGS
```

LABEL	VALUE	DEFN	REFERENCES
48BERSNMO	0008'	0066	0111
BERSYN	0004'	0065	0145 0154
BSYNCH	0000'	0064	0077
CBH63	R 0060'	0056	0110
CFI	R 006A'	0058	0113
CHAT	R 0098'	0058	0144
COMMA#	00B3	0061	0151 0153
DX10	0001	0003	0004 0070 0117
ENLN	R 002E'	0059	0094
ERRSYN	R 0006'	0056	0065
ERRT	R 000A'	0058	0066
EXTRAM	R 0056'	0056	0106
		0058	
FAC	R 0080'	0056	0098 0133
FAC10	R 006E'	0058	0112 0114
FAC2	R 005E'	0058	0110
GET1	R 0020'	0059	0090
GETG2	R 002A'	0059	0093
GETSTK	R 00BC'	0057	0157
GOTO90	R 0094'	0056	0141
LBLP#	00AB'	0152	0144
LPAR	0096'	0144	
LPAR#	00B7	0062	0152
P359	0000	0003	0003
PARCOM	009E'	0148	0079
PARSE	R 00AB'	0057	0150
PGMCHR	R 00B4'	0057	0155
PUTSTK	R 00A0'	0057	0148
R0	0000	0073	0076 0078 0080 0091 0114 0125 0126 0128 0129 0130 0137
R1	0001	0097	0098 0127 0128 0129 0130 0137
R10	000A	0089	0105 0107
R11	000B	0109	
R12	000C	0109	0139
R13	000D	0073	0125 0127 0129
R1LB	R	0056	
R2	0002	0133	0135 0137
R3	0003	0094	0095 0101 0102 0104 0106
R7	0007	0090	0093 0096
RB	0008	0153	
RAMFLG	R	0058	
RAMTOP	R 0024'	0059	0091
RANERR	008E'	0140	0115 0134 0136 0138
RANGE	005A'	0109	0081
RESET	R 00C0'	0057	0105 0158
SAVREG	R 00B8'	0057	0156
SEETW2	002C'	0094	0092
SEETW4	0034'	0096	0103
SEETW6	0050'	0105	0100
SEETW8	0054'	0106	0099
SEETW0	001A'	0089	
SET	R 001C'	0056	0089
SETREG	R 00A4'	0057	0140 0149
SPEED	D 000C'	0068	0055
STLN	R 004E'	0058	0102 0104
SYNCHK	R 0002'	0056	0064
VERMAC	M	A0001	0003
VERS	0000	0003	0004 0070 0117