

ACCESS NAMES TABLE

PAGE 0001

SOURCE ACCESS NAME= PPC2.P359.V081180.SRC.GETN8
OBJECT ACCESS NAME= PPC2.P359.V081180.OBJ.GETN8S
LISTING ACCESS NAME= PPC2.P359.V081180.LST.GETN8S
ERROR ACCESS NAME=
OPTIONS= KREF
MACHINE LIBRARY PATHNAME=

LINE	KEY	NAME
0002	A	VERSION =>PPC2.P359.V081180.SRC.P359

0031
0032
0033
0034
0035
0036
0037
0038
0039
0040
0041
0042
0043
0044
0045
0046
0047
0048
0049
0050

IDT 'GETNBS'

```
*****  
*  
*          GGGG          EEEEE          TTTTTT          N          N          BBBB  
*          G          E          T          NN          N          B  B  
*          G          E          T          N  N          N          B  B  
*          G  GG  EEE          T          N  N  N          BBBB  
*          G  G  E          T          N  N  N          B  B  
*          G  G  E          T          N          NN          B  B  
*          GGGG          EEEEE          T          N          N          BBBB  
*  
*          PPPP          3333          555555          9999  
*          P  P          3  3          5          9  9  
*          P  P          3  3          5          9  9  
*          PPPP          3333          55555          99999  
*          P          3  3          5  5          9  9  
*          P          3  3          5  5          9  9  
*          P          3333          5555          9999  
*  
*****
```

```

0055          DEF  GETNB,GETCHR
0056          REF  VARW,VARW1,VARA,VRAM,VDPD
0057 0000
0058          *      Get a non-space character
0059 0000 0008  GETNB  MOV  R11,RO      Save return address
0060 0002 08A0  GETNB1 BL   @GETCHR     Get next character
0061 0004 000E'
0061 0006 08B1          CI  R1, / (*256      Space character?
0061 0008 2000
0062 000A 13FB          JEQ  GETNB1      Yes - get next character
0063 000C 0450          B    *RO'      No - return char/condition
0064 000E
0065 000E
0066 000E
0067 000E
0068          *      Get the next character
0069 000E 8820  GETCHR C   @VARW,@VARA      End of line?
0069 0010 0000
0069 0012 0000
0070 0014 1D1E          JH   GETCH2      Yes - return condition
0071          *-----CONDITIONAL ASSEMBLY-----*
0072          ASMIF VERS=DX10
0073          MOV  @VARW,R14
0074          AI   R14,VRAM
0075          LI   R1,>A000
0076          AB  *R14,R1
0077          INC  @VARW      Increment read-from pointer
0078
0079          ASMELS
0080 0016
0081 0016 D7E0          MOVB @VARW1,*R15      No - write LSByte of VDP addr
0081 0018 0000
0082 001A 0201          LI   R1,>A000      Negative screen offset (->60)
0082 001C A000
0083 001E D7E0          MOVB @VARW,*R15      Write MSByte of VDP addr
0083 0020 0010'
0084 0022 05A0          INC  @VARW      Increment read-from pointer
0084 0024 0020'
0085 0026 B060          AB  @VDPD,R1      Read and remove screen offset
0085 0028 0000
0086          ASMEND
0087          *-----END OF CONDITIONAL ASSEMBLY-----*
0088 002A 0281          CI  R1,>1F00      Read an edge character?
0088 002C 1F00
0089 002E 13EF          JEQ  GETCHR      Yes - skip it
0089 0030 045B          RT      Return
0091 0032 04C1  GETCH2 CLR  R1      Indicate end of line
0092 0034 045B          RT      Return
0093          END
NO ERRORS,          NO WARNINGS

```

LABEL	VALUE	DEFN	REFERENCES
DRIC	0001	0003	0004 0072
GETOBE	0032	0091	0070
GETOHP	000E	0059	0055 0050 0089
GETYB	0000	0059	0055
GETERL	0012	0050	0022
PIBP	000E	0003	0002
R0	0010		0059 0023
R1	0011		0051 0082 0055 0088 0091
R11	0003		0059
R15	000F		0081 0083
VARA	R 0012	0056	0069
VARW	R 0014	0056	0069 0083 0084
VARW1	R 0018	0056	0081
VEPRB	R 0028	0056	0085
VERMAC	M	A0001	0003
VERE	0000	0003	0004 0072
VRAM	R	0056	