```
Declarations for Pass 3
 *** Compilers 31/7/76

    manifest GC = 430

    global
    §g
 5      Command             : GC
        Expression          : GC + 1
        LeftExpression      : GC + 2
        ConstantExpression  : GC + 3
        ConstOp             : GC + 4
10      Application         : GC + 5
        DealwithAss         : GC + 6
        FindName            : GC + 7
        Prec                : GC + 8
        StackOutput         : GC + 9
15      ClearStack          : GC + 10
        Addto               : GC + 11
        Subtractfrom        : GC + 12
        NormalOut           : GC + 13
        ChangeOutput        : GC + 14
20      DeleteCommand       : GC + 15
        CheckScope          : GC + 16
        SetUp3b             : GC + 17
        CloseDown3b         : GC + 18
        ManFunDefinition    : GC + 19
25      ManFunDefParams     : GC + 20

        NormalOutput        : GC + 21
        CellOutput          : GC + 22
        OutputVec           : GC + 23
30      ShuntVec            : GC + 24
        NonRecVec           : GC + 25
        NonRecDef           : GC + 26
        PresentOutput       : GC + 27
        FHN                 : GC + 28
35      LastRelExt          : GC + 29
        SSP                 : GC + 30
    $g

    manifest
40  §m
        BOTTOM     = 8366
        SUBEXP     = 8367

        NORMAL     = 0
45      MFAPP      = 1
        MFDEF      = 2
        CONSTANT   = 3
        IGNORE     = 4

50      SIZE       = 0
        PTR        = 1
        FIRSTEL    = 2

        SHUNTSIZE  = 50
55      DEPTHSIZE  = 20
        STACKSIZE  = 20
        BUFFERSIZE = 200
    $m

60  manifest
    §m
```

```
    UpdateSSP[T] is
        SSP := SSP + StackIncrement[T]
65
    Code[Type, n] is
    §C
        ClearStack[]
        NormalOutput[2, Type, n]
70      UpdateSSP[Type]
    §C

    Inst[Type] is
    §I
75      ClearStack[]
        NormalOutput[1, Type]
        UpdateSSP[Type]
    §I

80  ForHop[n] is
    §FH
        ClearStack[]
        NormalOutput[2, FORHOP, n]
    §FH
85
    ForHopPt[n] is
        if n > 0 do
        §   ClearStack[]
            NormalOutput[2, FORHOPPT, n]
90      §

    BackHop[n] is
    §BH
        ClearStack[]
95      NormalOutput[2, BACKHOP, n]
    §BH

    BackHopPt[n] is
    §BHP
100     ClearStack[]
        NormalOutput[2, BACKHOPPT, n]
    §BHP

    CondForHop[Type, n] is
105 §CFH
        ClearStack[]
        NormalOutput[3, CONDFORHOP, n, Type]
        UpdateSSP[Type]
    §CFH
110
    SetLitGlobal[No, Loc, Val] is
        NormalOut[4, SETLITGLOBAL, No, Loc, Val]

    SetLitStatic[No, Val] is
115     NormalOut[3, SETLITSTATIC, No, Val]

    SetDSegGlobal[No, Loc, l] is
        NormalOut[4, SETDSEGGLOBAL, No, Loc, l]

120 SetDSegStatic[No, l] is
        NormalOut[3, SETDSEGSTATIC, No, l]

    SetCSegGlobal[No, Loc, Id] is
    §SCG
125     ClearStack[]
        NormalOutput[4, SETCSEGGLOBAL, No, Loc, Id]
```

```
        §SCG

        SetCSegStatic[No, Id] is
130     §SCS
            ClearStack[]
            NormalOutput[3, SETCSEGSTATIC, No, Id]
        §SCS

135     SetManFun[No, Val] is
        §SMF
            ClearStack[]
            NormalOutput[3, SETMANFUN, No, Val]
        §SMF
140
        LoadManFun[No, Code] is
        §LMF
            ClearStack[]
            NormalOutput[3, LOADMANFUN, No, Code]
145         UpdateSSP[RAsm]
        §LMF

        LoadStatic[Type, No] is
        §LS
150         ClearStack[]
            NormalOutput[3, LOADSTATIC, Type, No]
            UpdateSSP[Type]
        §LS

155     LoadString[No] is
        §LS
            ClearStack[]
            NormalOutput[2, LOADSTRING, No]
            UpdateSSP[Nsm]
160     §LS

        UpdateS[n] is
        §US
            ClearStack[]
165         NormalOutput[2, Ssm, n]
            SSP := n
        §US

        StartOutputMode[Mode] is
170     §   Addto[OutputVec, Mode]
            ChangeOutput[]
        §

        EndOutputMode[] is
175     §   Subtractfrom[OutputVec]
            ChangeOutput[]
        §

        StartNonRecDef[] is
180     §   NonRecDef := true
            Addto[NonRecVec, Generation[PresentLevel]]
        §

        EndNonRecDef[Old] is
185     §   Subtractfrom[NonRecVec]
            NonRecDef := Old
        §

        Top[Vec] = Vec↓(Vec↓PTR)
190
        SwitchNo[] = valof
```

```
        §SWN
            SWN := SWN + 1
            resultis SWN
195     §SWN

        HopNo[] = valof
        §HN
            HN := HN + 1
200         resultis HN
        §HN

        UnusedHopNo[] = -HopNo[]

205     HopNoUsed[h] = (h ≥ 0)

        CellLength[p] = LengthField[p↓0]

        CellType[p] = p↓0
210
        CellNo[p] = p↓1

        CellNo2[p] = p↓2
    §m
215

****
```