Lab Exercise #14 -- Assembler Processing

A. Complete the exercises below.

1. The second page of this worksheet lists the source lines for a SPARC assembly language program. Fill in each blank in the column on the left of the page with the appropriate hexadecimal value (the value of the location counter before that source line is processed during Pass One).

Note: the assembler maintains a separate location counter for each segment. Use the notation "T+xxxx" or "D+xxxx" (where "T" and "D" refer to the text and data segments, and "xxxx" refers to a hexadecimal offset) to give the value of the location counter.

2. Complete the symbol table shown below, based on the processing which you performed in part (1). For each symbol, indicate its value (a specific constant or a segment plus a hexadecimal offset), whether its value is absolute or relocatable, and whether it is a local or global symbol.

symbol	value (segment+offset)	abs/rel	local/global

3. For each of the indicated source lines, fill in the object code (machine instruction or data value) which would generated during Pass Two and placed in the object code file. Give your answers using hexadecimal notation.

B. Assemble the source code file and check your work using the following commands:

```
<prompt> gcc -c ~cse320/Labs/lab14.source.s
<prompt> dis -d .data -t .text lab14.source.o
```

Alternatively, you can use the following command to generate an assembly listing:

```
<prompt> ~cse320/bin/assem -a ~cse320/Labs/lab14.source.s
```

If any of your responses are incorrect, re-work that section of the worksheet.

```
                .section ".text"
                .align   4
unpack:
                set      masks, %l7
                ld       [%l7+0], %o2
                ld       [%l7+4], %o3
                ld       [%l7+8], %o4
                and      %o1, %o2, %o2
                and      %o1, %o3, %o3
                and      %o1, %o4, %o4

                retl                        ! _____________________________
                nop

                .section ".data"
                .align   4
list:           .single  0r-64.0
                .single  0r+1.625
                .single  0r-1.3e-6
                .single  0r-100.0625
                .single  0r+12.6e+32

masks:          .word    0x80000000
                .word    0x7f800000
                .word    0x007fffff

fmt:            .asciz   "Number: %8.8x Fields: %8.8x %8.8x %8.8x\n"
                .align   4

SIZE            =        5

                .global  main
                .section ".text"
                .align   4
main:
                save     %sp, -96, %sp

                mov      0, %l0
                set      list, %l2
loop:
                cmp      %l0, SIZE
                bge      endloop            ! _____________________________
                nop

                sll      %l0, 2, %l1
                ld       [%l2+%l1], %o1
                call     unpack             ! _____________________________
                nop
                set      fmt, %o0
                call     printf             ! _____________________________
                nop

                inc      %l0
                ba       loop               ! _____________________________
                nop
endloop:

                ret                         ! _____________________________
                restore
```