

CSCI 51A/E MID TERM 2

5/16/95

1. Each question on this test adds some more code to a single assembly language program which contains the following lines of code. Lines containing a sequence of dots indicate that there is some more code in that location which is not shown and is irrelevant. Do not attempt to put any code in place of the dots, they are there as space fillers. The main program follows, and calls subroutine MYSUB which has three arguments, X, Y, and Z. X and Y will be passed on the stack by value, and Z will be passed by reference.

The subroutine must preserve the values in ALL registers, and return with the stack cleaned up, with all arguments removed from the stack. The subroutine will use registers A0, A1, A2D0, D1, D2, and D3. These registers must be saved and restored by the subroutine. The subroutine will also need to reserve room for 5 words on the stack for local variables.

```
X    DC.W    1
Y    DC.L    2
Z    DS.L    75
.....
```

1A. Add line(s) of code that push the address of the array Z on the stack.

1B. Add line(s) of code that push the value of argument Y on the stack.

1C. Add line(s) of code that push the value of argument X on the stack.

1D. Add line(s) of code that call subroutine MYSUB

There are some more lines of code in the main program here, and then the subroutine begins:

.....

MYSUB:

1E. Add code that reserves room for the 5 word length local variables on the stack. Use the link instruction with A6 for the link register.

1F. Add the line(s) of code that save the registers on the stack that are used by the subroutine. Use the MOVEM instruction. How many locations does this add to the stack? _____

1G. Draw a picture of the stack as exists at this point in the program's execution. (Draw it below, and show the stack location pointed to by the stack pointer SP and by A6.)

1H. Add the line(s) of code that move the arguments X into D0 and Y into D1. Assume that previous instructions in the subroutine (which are not shown) have pushed more stuff onto the stack, and that the stack pointer is now at an unknown position.

1I. Add the line(s) of code that move the VALUE of the FIFTH member of the array Z into D2. The fifth member of Z would correspond to Z(4) if subscripts start at zero, as in the C language.

More lines of code follow that accomplish the subroutine's task. These lines may use the stack.

```
.....  
.....
```

1J. Add the line(s) of code that restore the registers that are used by this subroutine. Assume that all stuff pushed by the hidden lines in the subroutine have now been popped back off, and the SP is where it belongs.

1K. Add the line(s) of code that reverse the operation of the link instruction and restore A6 to its original value.

1L. Add the line(s) of code that remove the arguments from the stack and move the return address to the proper point for returning to the main program.

1M. Add the line(s) of code that return to the main program.

2A. Assume bit zero of a register is the least significant bit. Write a line of code that will use either an AND, OR, or XOR to extract bits 3-5 of D0 and

then add a conditional branch instruction that will branch to LABEL1 if the result is not zero.

2B. Write a line of code that will use an AND, OR, or XOR to set bit 2 of D0 to a one, regardless of its current value.

2C. Write a line of code that uses an AND, OR, or XOR to flip each of the bits in bits 3-6 of D0 to a zero if it is currently a one, and to a one if it is currently a zero.

3. Show the results after the following instructions. Assume that all condition code bits are zero before each set of instructions.

3A. MOVE.L #\$12345678,D0
 ASL #4,D0 D0=_____

C bit = _____
N bit = _____
Z bit = _____
V bit = _____
X bit = _____

3B. MOVE.L #\$12345678,D0
 LSR.L #4,D0 D0=_____

C bit = _____
N bit = _____
Z bit = _____
V bit = _____
X bit = _____

3C. MOVE.L #\$12345678,D0
 ROR.L #4,D0 D0=_____

C bit = _____

N bit = _____

Z bit = _____

V bit = _____

X bit = _____

3D. Write the assembly instructions to multiply the (longword) number in D0 by 12, using only MOVE and shift instructions.

3E. Write the assembly instructions to divide the (longword) number in D0 by 8, using only shift instructions.

4. Show (in HEX) the IEEE single precision floating point representation for the decimal number 14.25