

<File: Exam #4 F'97>
EEAP 282 - FALL 1997

Exam #4
December 5, 1997

NAME: _____ CWRUnet ID: _____

SOLUTIONS

IMPORTANT INFORMATION:

Exam is closed book, closed notes. Only the M68000 Programmer's Reference Manual and/or Programming Reference Card are allowed to be used.

	Possible
1.	20
2.	10
3.	16
4.	10
5.	10
6.	25
SCORE	91

50 took exam

1. This program calls a subroutine using LINK and UNLK instructions. Assume that program execution starts at STRT and that (A1)=\$00004000 initially. What is on the stack:

- (a) just AFTER the instruction JSR is executed
- (b) just AFTER the instruction LINK is executed
- (c) just BEFORE the UNLK instruction
- (d) just AFTER the instruction RTS is executed

Line	Address				
1			ORG		\$1000
2	00000008	N	EQU		8
3	00000008	M	EQU		8
4					
5	00001000	DFFC FFFF FFF8	strt	ADD.L	#-N,SP
6	00001006	2F38 101E 4E71		MOVE.L	ARG,-(SP)
7	0000100C	4878 1022 4E71		PEA	X
8	00001012	4EB8 10EA 4E71		JSR	SUBR
9	00001018	508F		ADD.L	#8,SP
10	0000101A	221F		MOVE.L	(SP)+,D1
11	0000101C	241F		MOVE.L	(SP)+,D2
12	0000101E	0123 4567	ARG	DC.L	\$01234567
13	00001022		X	DS.B	200
14					
15	000010EA	4E51 FFF8	SUBR	LINK	A1,#-M
16	000010EE	2378 1112 FFFC 4E71		MOVE.L	LCL1,-4(A1)
17	000010F6	2378 1116 FFF8 4E71		MOVE.L	LCL2,-8(A1)
18	000010FE	52A9 FFFC		ADD.L	#1,-4(A1)
19	00001102	2469 0008		MOVEA.L	8(A1),A2
20	00001106	2378 111A 0010 4E71		MOVE.L	OUT1,16(A1)
21	0000110E	4E59		UNLK	A1
22	00001110	4E75		RTS	
23					
24	00001112	9876 5432	LCL1	DC.L	\$98765432
25	00001116	8765 4321	LCL2	DC.L	\$87654321
26	0000111A	4142 4344	OUT1	DC.L	'ABCD'
27					
28				END	strt

Symbol Table

Label	Value
ARG	0000101E
LCL1	00001112
LCL2	00001116
M	00000008
N	00000008
OUT1	0000111A
SUBR	000010EA
X	00001022
start1	00001000

(a)	(b)	(c)	(d)
	xxxx<--sp	8765<--sp	
	xxxx	4321	
	xxxx	9876	
	xxxx	5432+1=5433	
	0000<--a1, FP	xxxx<--a1,FP	
	4000	xxxx	
0000<--sp	0000	0000	
1018	1018	1018	
0000	0000	0000	0000<--sp
1022	1022	1022	1022
0123	0123	0123	0123
4567	4567	4567	4567
xxxx	xxxx	xxxx	4142
xxxx	xxxx	xxxx	4344
xxxx	xxxx	xxxx	xxxx
xxxx	xxxx	xxxx	xxxx

original sp -->

--	--	--	--

Be sure to indicate the contents of each memory location. If the contents are not known use "xxxx" to indicate an unknown state. The stacks are shown as **WORD** width.

Grading:

- 1 for contents instead of address
- 1 for upper and lower words reversed
- 2 for not reserving LINK space
- 2 for not putting space on stack from ADD #-N,SP
- 1 for not indicating frame pointer

2. You now are a TA for EEAP282. A new student is trying to do lab#4.

```
sine    movem.l  A0-A6/D0-D7,-(SP)  ;preserve prior state
        move.l  15(SP),A0           ;RA in A0
```

The debugger stops at the above code and displays the error message “exception vector 3” Explain what is wrong with this code. Be as explicit as possible.

ANSWER: exception vector 3 is an attempt to do a word or long word operation at an odd address. -3 points for just saying that it's an address error.

3. For (a)-(d) indicate if that privilege violation and, if not, describe the detailed effect of executing the instruction on both the status register and the state of the machine. Be as explicit as possible.

(a) `ANDI.W #0x70FF,SR` ;in the supervisor mode

(b) `EORI.W #0x8000,SR` 'in the supervisor mode

(c) `EORI.W #0x0011,SR` ;in the user mode

(d) `MOVE.W #0x2400,SR` 'in the user mode

Answer:

- (a) clears the trace bit, remains in supervisor mode, sets interrupts to %000, leaves the CCR unchanged
- (b) toggles the trace bit, mode and interrupts remain the same, leaves the CCR unchanged
- (c) toggles the X and C bits, everything else remains the same
- (d) technically a privilege violation so nothing changes. If it does change it would clear the trace bit, put the 68000 into supervisor mode*, sets the interrupt mask to level 4, clears all bits of the CCR.

*privilege violation

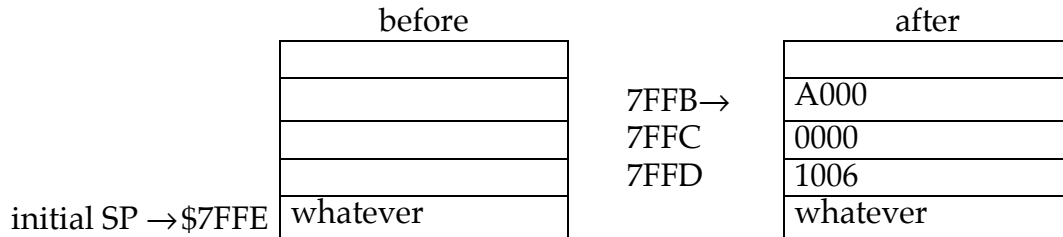
NOTE: The use of CCR instead of SR would disable transfers to the upper byte of the Status Register preventing privilege violations. This occurred in (c) and (d).

GRADING:

-4 points per section

4. The instruction TRAP #3 is located at location \$1004 and executes when (SR)=\$A000 and (SP)=\$7FFE.

- (a) Is the processor initially in user or supervisor mode?
- (b) Show the contents of the system stack just before and after the TRAP instruction executes.



- (c) Give the code to initialize the TRAP #3 vector so that the trap routine starts at \$1100.

Answer:

- (a) Supervisor mode (with trace on). -
- (b) See above
- (c) The vector number is 32 (\$80) plus the TRAP number, i.e. 32+3=35. The vector address is 4 times the vector number or $(35_{10}=\$23) \times 4$ which, in this case, is \$8C (140_{10}). So, a typical instruction to set the trap vector address would be:
 MOVE.L #\$1100,\$8C

5. The 68000 is in supervisor mode and executes the following program fragment.

```
MONITOR    EQU      $9000          ;start. address user program
           MOVE.L   #$3C00,-(SP)   ;load start address onto
system stack
           MOVE.W   #$8000,-(SP)   ;now load a SR which is
configured for user mode, trace on, interrupt level 0
           RTE      ;pop the SR and PC off the
system stack; start the program in user mode at PC=$9000
```

(a) What do the two MOVE instructions do? A picture of the appropriate stack is expected.

ANSWER: Load the system stack with

```
SP--->[$8000]
        [$0000]
        [$3C00]
```

(b) What does the RTE instruction do? Be explicit as to the state of the machine.

ANSWER: pops the SR=\$8000 off the stack and starts program execution at \$3C00 with the TRACE turned off in user mode.

GRADING: 5 points per part.

6. You are doing a new lab (which will be assigned next year) where you must transmit a block of data (a network packet) using a TRAP call.

(a) Give explicit 68000 code which will make the 68000 execute your code which begins at \$2800 in response to the TRAP #5 call shown in the above program fragment. Be sure to show the calculation of vector number, address, etc.

ANSWER: The exception vector number is $32+5=37$. According to the rules for exceptions, the exception vector table address is then $37 \times 4 = 148 = \$94$. You would then need to put the following instruction (or something equivalent) near the beginning of your program:

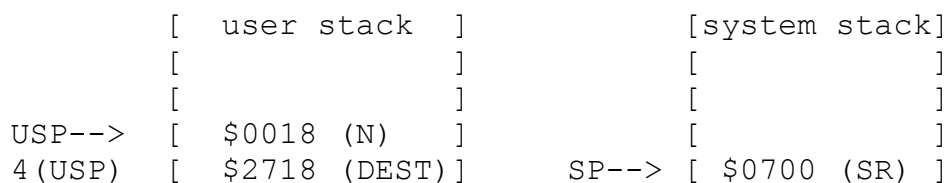
```
ANS  MOVE.L    #$2800,$94 ;load location of
                                ;exception handling routine
```

Scoring: 7 points total. -2 points if no instruction like the above. -5 points if incorrect address. Three points for coming up with the vector number; an additional two points for the exact address, i.e. \$94.

For the following sections, assume that the status register is initially \$0700 and that you are executing your packet transmission program in the debugger. You start executing your routine in response to the TRAP call.

(b) Draw a picture of the system and user stacks at the point labeled STACKS: in your TRAP #5 packet transmission routine.

ANSWER:



8 (USP)	[\$0000 (BUF)]	[\$0000 (RA)]
	[\$2700]	[\$2510]

Note: these stacks are shown as word width instead of the usual byte width for brevity. Grading 5 points for user stack, 3 points for system stack

- (c) Write explicit 68000 instructions to retrieve the address of the message which is to be transmitted. These instructions should put the starting address of the packet into A0, the number of bytes in the packet into D0, and the destination address into D1. These instructions should be located in the program after the label PARAMETERS.

ANSWER

```

MOVE.L    USP,A2          ;get the value of the USP
                          ;and put it into A2
MOVE.W    (A2)+,D0        ;get number of bytes
MOVE.W    (A2)+,D1        ;get destination
MOVEA.L   (A2),A0         ;get location of message

```

Alternative answer:

```

MOVE.L    USP,A2          ;get the value of the USP
                          ;and put it into A2
MOVE.W    (A2),D0         ;get number of bytes
MOVE.W    2(A2),D1        ;get destination
MOVEA.L   4(A2),A0        ;get location of message

```

GRADING. Tough to grade. 10 points total. Answer must use USP. 10 points total. Lots of possible answers.

CODE for Problem 6:

```
                ORG      $2500
* put input parameters on stack
START:         MOVE.L   #BUF,-(SP) ;starting address of data
                ;to transmit
                MOVE.W  #DEST,-(SP);destination address
                MOVE.W  #N,-(SP)  ;number of bytes
                TRAP    #5         ;call your send routine
                MOVE.W  (SP)+,STAT ;get status of transmission

                ORG      $2700
BUF            DC.B
'1','2','3','4','5','6','7','8'
                DC.B
'A','B','C','D','E','F','G','H'
                DC.B
'!','@','#','$','%','^','&','*'
DEST          DC.W      $A001      ;destination
STAT          DS.W      1
N             EQU      24

                ORG      $2800
STACKS:
PARAMETERS:
; YOUR PACKET TRANSMISSION CODE WOULD GO HERE
END
```

FOR YOUR REFERENCE THE EXCEPTION VECTOR TABLE IS:

vector number (Decimal)	address (Hex)	assignment
0	0000	RESET: initial supervisor stack pointer (SSP)
1	0004	RESET: initial program counter (PC)
2	0008	bus error
3	000C	address error
4	0010	illegal instruction
5	0014	zero divide
6	0018	CHK instruction
7	001C	TRAPV instruction
8	0020	priviledge violation
9	0024	trace
10	0028	1010 instruction trap
11	002C	1111 instruction trap
12*	0030	not assigned, reserved by Motorola
13*	0034	not assigned, reserved by Motorola
14*	0038	not assigned, reserved by Motorola
15	003C	uninitialized interrupt vector
16-23*	0040-005F	not assigned, reserved by Motorola
24	0060	spurious interrupt
25	0064	Level 1 interrupt autovector
26	0068	Level 2 interrupt autovector
27	006C	Level 3 interrupt autovector
28	0070	Level 4 interrupt autovector
29	0074	Level 5 interrupt autovector
30	0078	Level 6 interrupt autovector
31	007C	Level 7 interrupt autovector
32-47	0080-00BF	TRAP instruction vectors**
48-63	00C0-00FF	not assigned, reserved by Motorola
64-255	0100-03FF	user interrupt vectors

NOTES:

* No peripheral devices should be assigned these numbers

The following logic functions may be needed at various points throughout the exam.

A	B	A OR B	A AND B	A EOR B
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

** TRAP #N uses vector number 32+N

Name: _____ CWRUnet ID: _____ Page 12

HINT: The debugger subroutine PutString will output to the debugger widow the sequence of ASCII bytes which begin at the address in A0. PutString will continue outputting these ASCII charaters to the screen until it encounters a zero byte, i.e. \$00, at which point it stops.