Appendix B

MC68000 Review Questions & Programs

by

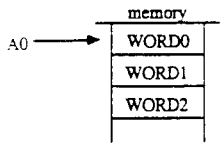
Motorola Inc .

Semiconductor Products Sector

PROGRAM PROBLEM 1

Write a program to ADD together 10 words of DATA. This DATA is STORED in SEQUENTIAL memory in ASCENDING ADDRESS LOCATIONS. The long word RESULT should be placed into D0. A0 points to the FIRST word in the STRING.

WORD0 + WORD1 + • • • • + WORD9 ⟶ D0



MT8-66[illegible]

PROGRAM PROBLEM 2

Write a subroutine to clear memory from A0 (passed in) thru A1 (passed in) and A1 ≥ A0.

MT8-66[illegible]

PROGRAM PROBLEM 3

Write a subroutine to find Z. Where: $Z = (X - Y)^2$

AND $0_{10} \leq Y \leq X \leq 32{,}000_{10}$

D0 contains Y, D1 contains X and D2 will contain Z.

MT8-663.2

DAY 1 REVIEW

. Within the 68000 there are_____ address register(s), _____ data register(s),_____ stack pointer(s), _____ program counter(s), and _____status register(s).

2. The user stack pointer is called _____ or _____; the supervisor stack pointer is called _____ or _____.

3. The supervisor mode is indicated internally by the _____ and externally by_____ .

4. The most significant byte of a word is accessed on an _____ byte address.

5. What is the minimum time for: a) a read BUS cycle?_____ b) a write BUS cycle? _____ c) What is the maximum time?_____

6. What is the state of FC0 and FC1 for any write BUS cycle? _____.

7. When using the post-increment addressing mode, the address register is incremented by _____, _____, or _____, depending on whether the instruction is _____, _____, or _____.

8. In executing a branch instruction, the 68000 calculates the address by adding the displacement to _____.

9. The destination for a DIVU, DIVS, MULU, or MULS must always be a _____ .

10. Prior to the execution of the instruction

EXT.W D0

the register D0 contains $FFFFFF5C. The contents after execution of the instruction will be_____ .

DAY 2 PINS REVIEW

1. The _____ pin can be either an input or an output. It is an output when a double bus fault occurs.

2 If the processor is halted, what is the state of:

Address Bus _____
Data Bus _____
Control signals_____

3. The _____ can be used to indicate to the 68000 that a _____ is overdue.

4. If there are no interrupt requests pending, the interrupt pins will be _____.

5. In order to use auto-vectors, the _____ pin must go low during the interrupt acknowledge cycle.

6. If a level of 6 is in the status mask of the 68000, what interrupt levels will be allowed to be serviced? _____

7 Assume the 68000 responded to a level of 7 interrupt. During exception processing for that interrupt, the level continues to be asserted while a seven is moved into the interrupt mask. Will the 68000 respond again to a level 7. Why or Why not? _____
_____
_____

DAY 2
INSTRUCTION SET AND ADDRESSING MODE REVIEW

1. When using the address register indirect with displacement addressing mode, the size of the displacement is _____.

2. Indexing can be done with (circle the correct answer(s)):
   a. A data register only.
   b. An address register only.
   c. Either a data or address register
   d. A memory location

3. The size of an index register can be a (circle the correct answer(s)):
   a. Byte
   b. Word
   c. Long word

4. When using the address register indirect with index addressing mode, the size of the displacement is _____.

5. The difference between an ADDQ and an ADDI is that _____
   _____
   _____.

6. Rotate and shift instructions can only be used on _____
   and _____.

7. The compare instructions affect all the condition code bits except _____.

8. The instruction which can be used to initialize the user stack pointer from the supervisor mode is _____.

9. The instruction ROR D1, D3 the contents of _____
   will be rotated by the number of times in _____.

10. The RTR and RTE instructions are the same except that _____
    _____.

11. A STOP instruction is terminated if _____
    or _____
    or _____.

DAY 2
EXCEPTION PROCESSING REVIEW

1. If the STOP or RESET instructions are executed in user mode, a _____ exception occurs.

2. If a bus error occurs during an interrupt acknowledge, a _____
   _____ exception occurs.

3. An _____ exception occurs when an instruction attempts to access a word on an odd boundary.

4. A double bus fault occurs when:
   a. _____
   b. _____
   c. _____

5. A part of all exception processing except RESET is to store the _____ and _____ to the stack.

6.

ADDRESS	OPCODE	INSTRUCTION
$2000	$39C14000	_____

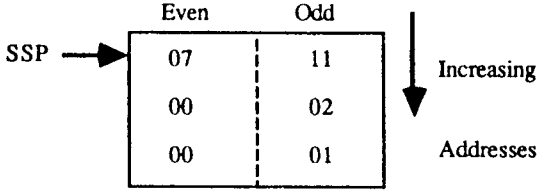
   If the above instruction is assembled and executed at address $2000 in memory, what kind of exception occurs? _____

7. During the exception processing sequence for the above example, what is the value of the PC saved on the supervisor stack? _____

8. The bit used to indicate that an exception is to occur after the execution of each instruction is _____.

9. The number $21 is read in from an interrupting device during exception processing. The vector for this is _____.

10.



This system will go into exception processing shortly after an RTE instruction because
_____
_____
_____

DAY 3
PROGRAM EXAMPLE PROBLEMS

1. Write the instructions necessary to replace

   LINK A3,#-$10

2. Write the instructions necessary to replace

   UNLK A3

3. Describe the results of executing the following program:

```
        LEA     $3000,A0
        LEA     $4000,A7
        MOVE.W  #2,D0
AGAIN   MOVE.B  (A0)+,(A7)+
        DBF     D0,AGAIN
```

4. The following program will move_____ bytes of data:

```
        MOVE.W  #30,D0
AGAIN   MOVE.W  0(A0,D0), 0(A1,D0)
        DBF     D0,AGAIN
```


LAB DAY 2

1. Fill in all missing information.

2. Figure out what each instruction does to the appropriate register(s).

3. If the program loops, how many times does it loop?_____.

4. What is the BLT instruction testing? (Hint: Which instruction affected the condition codes last?) _____

   _____.

5. What is the first address that the instruction @ 2016 writes to? _____.

6. What is the first address that the instruction @ 2020 writes to? _____.

7. Verify your answers to 3, 4, 5, and 6 by running the program in the ECB module.

LAB FOR DAY 2

```
002000  307C1000        MOVEA.W   _ _ _ _ _ _,_ _
002004  _ _ _ _00F0      MOVE.W    #$00F0,D2
002008  363C0210        MOVE.W    #$0210,D3
00200C  303CFFFA        MOVE.W    #$FFFA,D0
002010  _ _ _ _   LOOP  MOVE.W    (A0),D1
002012  E219            ROR._     _ _,_ _
002014  65_ _           BCS       ODD
002016  31A8_ _ _ _ _ _ _ _      MOVE._    00(A0),$10(A0,D2.W)
00201C  5543            SUBQ.W    #$2,D3
00201E  6006            BRA       _ _ _ _
002020  319030F0  ODD   MOVE.W    (A0),_ _ _ _ _ _ _ _
002024  _ _ _ _         SUBQ.W    #$2,D2
002026  5240      TEST _ _ _ _,_ _      #$1,D0
002028  5448            ADDQ.W    #$2,A0
00202A  6D_ _           BLT       LOOP
00202C  6000FFFE  END1  BRA       END1
```


LAB DAY 2: USING THE MC68KECB/CONSOLE SYSTEM

]1) The prompt TUTOR 1.X> should be displayed on console.

2) Type after prompt **MM 2000;W** and wait for response. Tutor will display the present WORD found at $2000 in memory and prompt you with "?" for you to enter 307C (the sequence of words beginning at address $2000). After typing **307C** (no $ required) then depress return key on console. This will open up next word location in memory, i.e. 2002?. Continue until you have entered all the program data.

NOTE: The memory modify, MM command of tutor has several subcommands:

(<DATA>) (CR)	UPDATE LOCATION AND SEQUENCE FORWARD
(<DATA>) ^ (CR)	UPDATE LOCATION AND SEQUENCE BACKWARD
(<DATA>) = (CR)	UPDATE LOCATION AND REOPEN SAME LOCATION
(<DATA>) . (CR)	UPDATE LOCATION AND TERMINATE

Where (<DATA>) is the data and (CR) is the return key.

3) Now type **MD 2000 2E;DI**. Compare result displayed with your worksheet.

4) Initialize the PC by typing after prompt **.PC 2000** then return key on console.

5) Next type **MM 1000;W** and wait for response. Then enter after "?" prompt:

0001 (CR)
0002 (CR)
0003 (CR)
0004 (CR)
0005 (CR)
0006 (CR)
0007 (CR)
0008.(CR)

Where (CR) represents depressing the return key on the console.

This sequence of entries will put some data (words of data) into memory at address $1000.

6) Type **DF** and the return key.

7) Type **TR** and return key on console. This will cause the display format to be printed on console. Then a new prompt will be displayed ":". Depressing the console's return key will cause the next instruction to be traced.

8) Continue to depress **return** to trace each individual instruction, one at a time.

9) Does the displayed information agree with your predicted results?

LAB FOR DAY 4

Search a string1 for all words which are evenly divisible by five. As a result of this search, create a string2 of all words examined which are evenly divisible by five. Create a string3 of all the words examined which are not evenly divisible by five.

Upon entering this routine, A0 is pointing to the beginning of string1, D0 contains the number of words in string1, A1 is pointing to the beginning of string2, and A2 is pointing to the beginning of string3.

The solution should be written for the general problem. However for the purposes of the lab session, string1 begins at address $4000 and is 8 words long. String2 starts at address $4100, and string3 starts at address $4200.

LAB DAY 4 (ECB ONLY)

I. HOW TO ENTER APROGRAM:

Type the following command: **MM 2000;DI**

This command means "Modify Memory starting at address $2000 in the DIsassembly mode.
This mode allows source code to be entered into memory line by line. Labels cannot be used. This requires you to ALWAYS press the space key before entering the instruction on each new line.

If the line by line assembler returns the same address followed by "X?" this means it didn't understand your input and it's giving you another chance (possibly you forgot the space?)

To enter branch instructions, since labels cannot be used, type the address where the MPU is to go. The displacement will automatically be calculated by the one-line assembler. For forward branches to instructions not yet entered, an asterisk can be used in place of the unknown address; and then once you do know the address, you can go back and retype the branch instruction with the known address (don't forget the dollar sign before the address).

II. HOW TO GET A LISTING OF YOUR PROGRAM:

Use the memory Display (MD) command.

MD 2000 34;DI

The **2000** is the starting address to display; the 34 is the number of bytes to display; the **;DI** is the DIsassembly mode.

III. HOW TO INSERT AN INSTRUCTION:

The Block Move (BM) command can be used to copy existing information in memory to a new location. You may wish to do this to make room to insert new instrucitons.

BM 2010 2020 2016 <= This command would move the memory contents between 2010 and 2020 up to memory locations starting at 2016. NOTE. Once this is done, check to see if the branches go to the correct locations now

IV. HOW TO CHANGE THE CONTENTS OF AN MPU REGISTER:

Type a period followed by the register designation, then a space, then the value.

.PC 2000 <=This changes to program counter to 0002000.
.D4 1000 <= This changes data register number 4 to 0001000.

V. HOW TO RUN YOUR PROGRAM:

Just type **GO**.
"GO" runs your program from the present value of the program counter. When you would like to continue from a breakpoint, just type "GO" again.

VI. HOW TO SET AND REMOVE BREAKPOINTS:

BR 2024 <= This sets a breakpoint at address 2024 which must be the location of an opword.
NOBR <= This removes all breakpoints.
NOBR 2024 <= This removes the breakpoint set at address 2024.

ECB COMMANDS

BF <address1> <address2> <word>	block of memory fill
BM <address1> <address2> <address3>	block of memory move
BR [<address>[;<count>]]	breakpoint set
BS <address1> <address2> <data>[<mask>][;option]	block of memory search
BT <address1> <address2>	block of memory test
DC <expression>	data conversion
DF	display formatted registers
DU [<port number>] <address1> <address2> [<text..>]	dump memory (S records)
GD [<address>]	go direct
GO [<address>]	go
GT <breakpoint address>	go until breakpoint
HE	help
LO [<port number>][;<options>][=text]	load (S records)
MD <address> [<count>][;<options>]	memory display
MM <address>[;<options>]	memory modify
MS <address> <data ...>	memory set
NOBR [<address> <address> ...]	breakpoint remove
NOPA	reset printer attach
OF	display offsets
PA	printer attach
PF[<port number>]	port format
RM	register modify
TM [<exit character>]	transparent mode
TR [<count>]	trace
TT <breakpoint address>	temporary breakpoint trace
VE [=text}	verify (S records)
* text ...	send message to port 2
.A0 - .A7 [<expression>]	display/set address register
.D0 - .D7 [<expression>]	display/set data register
.R0 - .R6 [<expression>]	display/set offset register
.PC [<expression>]	display/set program counter
.SR [<expression>]	display/set status register
.SS [<expression>]	display/set supervisor stack pointer
.US [<expression>]	display/set user stack pointer
(break)	abort command
(del)	delete character
(ctrl-D)	redisplay line
(ctrl-H)	delete character
(ctrl-W)	suspend output (1)
(ctrl-X)	cancel command line
(cr)	send line to memory

note: (1) when ctrl-W is used, the output display can be continued by entering any character.

MT8-671-1

Appendix C

Sloution to Review Questions & Programs

by

Motorola Inc .

Semiconductor Products Sector

MC68000 – ADDRESS AND DATA REGISTER DIFFERENCES

	DATA REGISTER	ADDRESS REGISTER
CCR	updated	not affected
byte operands	only bits 0-7 used and affected bits 8-31 unused and unaffected	not allowed
word operands	only bits 0-15 used and affected bits 16-31 unused and unaffected	If An is the source: bits 0-15 used If An is destination: word is sign extended to longword and all 32 bits of An is used and affected
longword operands	all 32 bits used and affected	all 32 bits used and affected

MT8ANS- 510-2

DAY 1 REVIEW

1. Within the 68000 there are 7 or 8 or 9 or 10 address register(s), 8 data register(s), 2 or 8 or 2 stack pointer(s) 1 program counter(s), and 1 status register(s).
2. The user stack pointer is called USP or A7 ; the supervisor stack pointer is called SSP or A7 .
3. The supervisor mode is indicated internally by the S bit in status register and externally by FC2 .
4. The most significant byte of a word is accessed on an even byte address.
5. What is the minimum time for: a) a read BUS cycle? 4 clock cycles b) a write BUS cycle? 4 clock cycles c) What is the maximum time? infinity
6. What is the state of FC0 and FC1 for any write BUS cycle? FC0=1, FC1=0
7. When using the post-increment addressing mode, the address register is incremented by 1 , 2 , or 4 , depending on whether the instruction is byte , word , or longword .
8. In executing a branch instruction, the 68000 calculates the address by adding the displacement to opword location plus two .
9. The destination for a DIVU, DIVS, MULU, or MULS must always be a data register .
10. Prior to the execution of the instruction

EXT W D0

the register D0 contains $FFFFF5C. The contents after execution of the instruction will be $FFFF005C .

MT8ANS 705-4

DAY 2 PINS REVIEW

1. The $\overline{\text{HALT}}$ pin can be either an input or an output. It is an output when a double bus fault occurs.

2. If the processor is halted, what is the state of:
   Address Bus Three — Stated
   Data Bus Three — Stated
   Control signals Driven

3. The $\overline{\text{BERR}}$ can be used to indicate to the 68000 that a $\overline{\text{DTACK}}$ is overdue.

4. If there are no interrupt requests pending, the interrupt pins will be all Hi, all negated, or all not asserted.

5. In order to use auto-vectors, the $\overline{\text{VPA}}$ pin must go low during the interrupt acknowledge cycle.

6. If a level of 6 is in the status mask of the 68000, what interrupt levels will be allowed to be serviced ? 7

7. Assume the 68000 responded to a level of 7 interrupt. During exception processing for that interrupt, the level continues to be asserted while a seven is moved into the interrupt mask. Will the 68000 respond again to a level 7? Why or Why not? No, logic level on $\overline{\text{IPLX}}$ pins must change for new level 7.

DAY 2 INSTRUCTION SET AND ADDRESSING MODE REVIEW

1. When using the address register indirect with displacement addressing mode, the size of the displacement is 16 bits.

2. Indexing can be done with (circle the correct answer(s)):
   a. A data register only.
   b. An address register only.
   (c) Either a data or address register
   d. A memory location

3. The size of an index register can be a (circle the correct answer(s)):
   a. Byte
   (b) Word
   (c) Long word

4. When using the address register indirect, with index addressing mode, the size of the displacement is a byte (8 bits)

5. The difference between an ADDQ and an ADDI is that: 1. For ADDQ the data is part of the opword. 2. For ADDQ, the data range is 1 to 8. 3. ADDQ will operate on address registers.

6. Logic and shift instructions can be used only on data registers and memory.

7. The compare instructions affect all the condition code bits except the X bit.

8. The instruction which can be used to initialize the user stack pointer from the supervisor mode is MOVE.L USP.

9. In the instruction ROR D1, D3 the contents of D3 will be rotated by the number of times in D1, MOD 64.

10. The RTR and RTE instructions are the same except that 1. RTE is privileged. 2. The system byte of the status register is not affected by RTR.

11. A STOP instruction is terminated if 1. An allowed interrupt occurs, or 2. A hardware reset occurs, or 3. The trace bit is enabled prior to stop ins

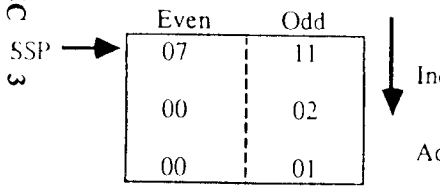
DAY 2
EXCEPTION PROCESSING REVIEW

1. If the STOP or RESET instructions are executed in user mode, a PRIVILIGE VIOLATION exception occurs.
2. If a bus error occurs during an interrupt acknowledge, a SPURIOUS INTERRUPT exception occurs.
3. An ILLEGAL ADDRESS ERROR exception occurs when an instruction attempts to access a word on an odd boundary.
4. A double bus fault occurs when:
   a. BERR DURING BERR EXCEPTION
   b. ANY COMBINATION OF BERR AND ILLEGAL ADDRESS
   c. BERR OR ILLEGAL ADDRESS DURING RESET
   d. ILLEGAL ADDRESS DURING ILLEGAL ADDRESS EXCEPTION
5. A part of all exception processing except RESET is to store the PROGRAM COUNTER and STATUS REGISTER to the stack.
6.

ADDRESS	OPCODE	INSTRUCTION
$2000	$39C14000	MOVE.W D1,#$4000

   If the above instruction is assembled and executed at address $2000 in memory, what kind of exception occurs? ILLEGAL INSTRUCTION
7. During the exception processing sequence for the above example, what is the value of the PC saved on the supervisor stack? $2000
8. The bit used to indicate that an exception is to occur after the execution of each instruction is THE T BIT IN THE STATUS REGISTER

The number $21 is read in from an interrupting device during exception processing. The vector for this is TRAP 1 AT ADDRESS $84



This system will go into exception processing shortly after an RTE instruction because OF AN ODD NUMBER ADDRESS FOR THE NEXT OPWORD READ BUS CYCLE

DAY 3
PROGRAM EXAMPLE PROBLEMS

1. WRITE THE INSTRUCTIONS NECESSARY TO REPLACE

```
LINK A3,# – $10
```

MOVE.L A3, – (A7)		PEA	(A3)
MOVEA.L A7, A3		LEA	(A7), A3
ADDA.L #-$10,A7		LEA	-$10(A7), A7

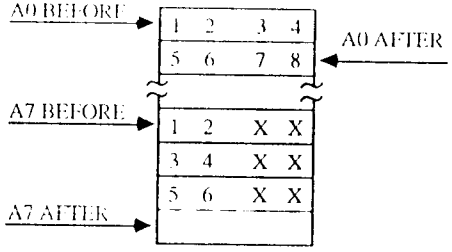
2. WRITE THE INSTRUCTIONS NECESSARY TO REPLACE

```
UNLK  A3
```

MOVEA.L A3, A7	LEA	(A3), A7
MOVEA.L (A7)+, A3	MOVEA.L	(A7)+, A3

3. DESCRIBE THE RESULTS OF EXECUTING THE FOLLOWING PROGRAM:

```
        LEA $3000, A0
        LEA $4000, A7
        MOVE.W #2, D0
AGAIN   MOVE.B (A0)+, (A7)+
        DBF D0,AGAIN
```



4. THE FOLLOWING PROGRAM WILL MOVE 0 or 2 BYTES OF DATA:

```
        MOVE.W #30, D0
AGAIN   MOVE.W 0(A0,D0), 0(A1,D0)
        DBRA D0, AGAIN
```

```
FIX –   MOVE.W #30,D0
        LEA +2(A0,D0),A0
        LEA +2(A1,D0),A1
AGAIN   MOVE.W –(A0), –(A1)
        DBF D0, AGAIN
```

- ILLEGAL ADDRESS DURING SECOND LOOP OR FIRST LOOP IF A0 OR A1 IS ODD

PROGRAM PROBLEM 1

WRITE A PROGRAM TO ADD TOGETHER 10 WORDS OF DATA. THIS DATA IS STORED IN SEQUENTIAL MEMORY IN ASCENDING ADDRESS LOCATIONS. THE LONG WORD RESULT SHOULD BE PLACED IN D0. A0 POINTS TO THE FIRST WORD IN THE STRING.

WORD0 + WORD1 + • • + WORD9 D0

A0

```
            SOLUTION 1          (UNSIGNED NUMBERS)
            ORG $2000
            CLR.L D0
            CLR.L D1
            MOVE.B #$09,D2
LOOP        MOVE.W (A0)+,D1
            ADD.L D1,D0
            SUB.B #$01,D2
            BHS LOOP
            END

            SOLUTION 2          (ASSUME SIGNED NUMBERS)
            CLR.L D0
            MOVE.B #$0A,D2
LOOP1       MOVE.W (A0)+,D1
            EXT.L D1
            ADD.L D1,D0
            SUB.B #$01,D2
            BNE LOOP1
            END

            SOLUTION 3          (UNSIGNED NUMBERS)
            MOVE.L A0,A1
            ADD.L #$14,A1
            CLR.L D0
            CLR.L D1
LOOP2       MOVE.W (A0)+,D1
            ADD.L D1,D0
            CMP.L A1,A0
            BLO LOOP2
            END
```

MT8:730-3

PROGRAM PROBLEM 2

WRITE A SUBROUTINE TO CLEAR (WRITE ZEROS) MEMORY FROM A0 (PASSED IN) THRU A1 (PASSED IN) AND A1 IS GREATER THAN OR EQUAL TO A0.

```
            SOLUTION 1          (NOTHING ASSUMED)
            ORG $2000
            CLR.B D0
START       MOVE.B D0,(A0)+
            CMP.L A0,A1
            BHS START
            RTS
            END

            SOLUTION 2
            CLR.W D0
LOOP        MOVE.W D0,(A0)+     (ASSUME A0 & A1 CONTAIN
            CMP.L A0,A1          EVEN ADDRESSES)
            BHS LOOP
            RTS
            END

            SOLUTION 3
BEGIN       CLR.L (A0)+         (ASSUME A0 & A1 CONTAIN
            CMP.L A1,A0          EVEN ADDRESSES AND A1 – A0
            BLS BEGIN            IS DIVISIBLE BY 4 WITH NO
            RTS                  REMAINDER)
            END
```

MT8:735-2

LAB DAY 2

1. Fill in all missing information.

2. Figure out what each instruction does to the appropriate register(s).

3. If the program loops, how many times does it loop? 6.

4. What is the BLT instruction testing? (Hint: Which instruction affected the condition codes last?) Instruction on line 2026 affected condition code last. BLT is testing to see if the value in D0 is still negative, and if so branch back to LOOP ([N ⊕ V=1]).

5. What is the first address that the instruction @ 2016 writes to? $1100.

6. What is the first address that the instruction @ 2020 writes to? $1200.

7. Verify your answers to 3, 4, 5, and 6 by running the program in the ECB module.

MT8ANS-664-1

PROGRAM PROBLEM 3

- IN THIS SPECIFIC CASE; $0 \le Y \le 32000$,
- $0 \le X \le 32000$, X AND Y ARE WORD SIZE,
- $Y \le X$ AND (X-Y) ALWAYS YIELDS A POSITIVE ANS.

SOLUTION 1

```
ENTER   MOVE.W   D1,D2     TO SAVE D1
        SUB.W    D0,D2      (X-Y)
        MULU     D2,D2      ABSOLUTE VALUE IN D2
        RTS
```

- IN A MORE GENERAL CASE, THE RESTRICTION
- $Y \le X$ NEED NOT APPLY AND ANY POSSIBLE
- COMBINATION OF VALUES MAY BE USED
- THEN, NOT ONLY OVERFLOW, BUT ALSO A
- NEGATIVE RESULT FROM (X-Y) MIGHT
- OCCUR. THIS CAN BE RESOLVED BY ADDING
- A SIGNED MULTIPLY TO THE SOLUTION ABOVE
- AND TESTING THE RESULT OF THE (X-Y)
- ARITHMETIC.

SOLUTION 2

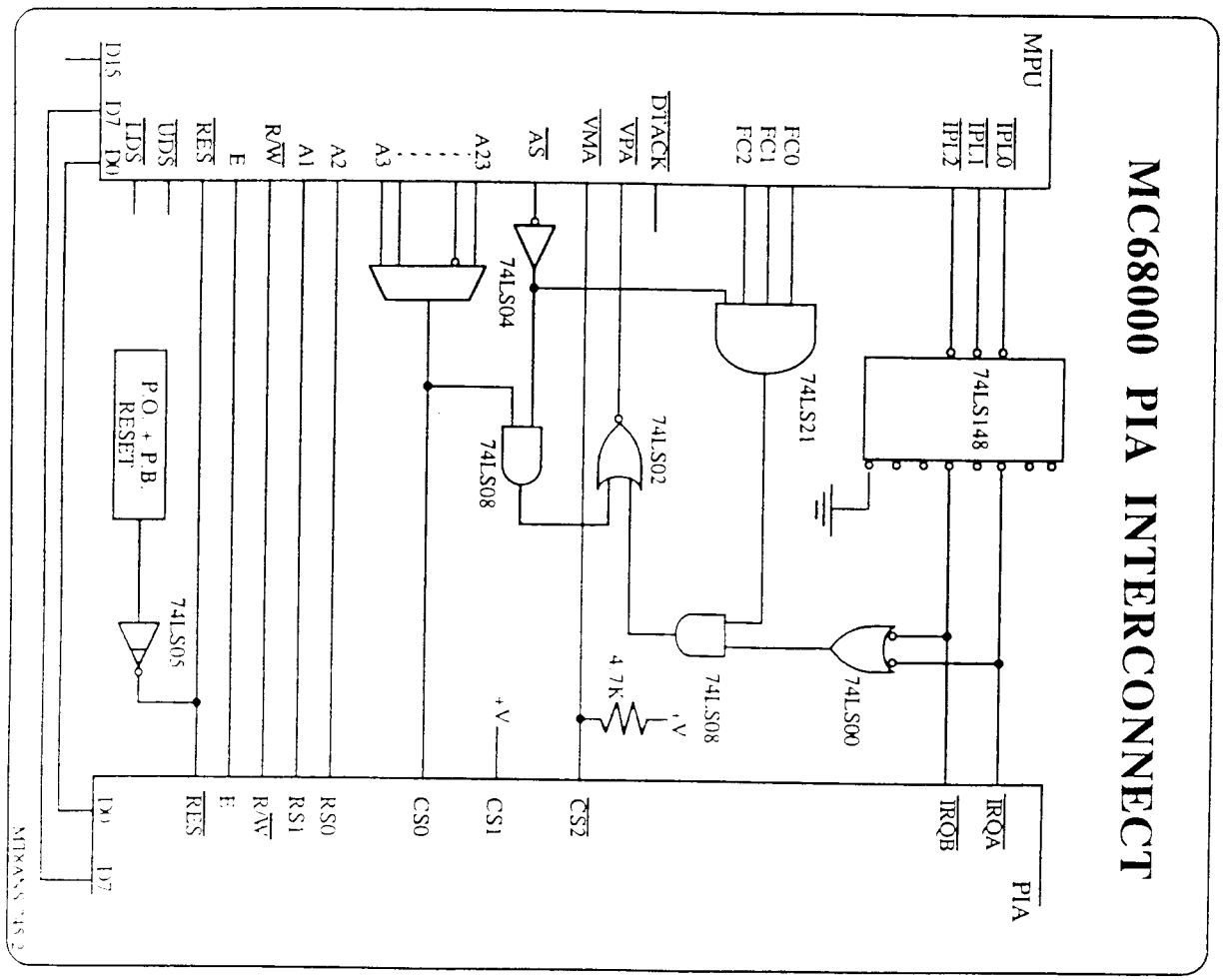
```
ENTER     MOVE.W   D1,D2     TO SAVE D1
          SUB.W    D0,D2      (X-Y)
          BVS      UNSIGNED   OVERFLOW?
          BMI      SIGNED     NEGATIVE?
UNSIGNED  MULU     D2,D2      SQUARED+OR ABSOLUTE (V=1)
          RTS
SIGNED    MULS     D2,D2      SQUARED
          RTS
```

MT8-740-2

ANSWER TO LAB FOR DAY 2

002000	307C1000		MOVEA.W	#$1000,A0
002004	343C00F0		MOVE.W	#$00F0,D2
002008	363C0210		MOVE.W	#$0210,D3
00200C	303CFFFA		MOVE.W	#$FFFA,D0
002010	3210	LOOP	MOVE.W	(A0),D1
002012	E219		ROR.B	#1,D1
002014	650A		BCS	ODD
002016	31A800002010		MOVE.W	00(A0),$10(A0,D2.W)
00201C	5543		SUBQ.W	#$2,D3
00201E	6006		BRA	TEST
002020	319030F0	ODD	MOVE.W	(A0),-$10(A0,D3.W)
002024	5542		SUBQ.W	#$2,D2
002026	5240	TEST	ADDQ.W	#$1,D0
002028	5448		ADDQ.W	#$2,A0
00202A	6DE4		BLT	LOOP
00202C	6000FFFE	END1	BRA	END1

MT8ANS-684



SOLUTION TO LAB DAY 4

```
*************************************
** SOLUTION 1 (SIGNED NUMBERS)   **
*************************************

        ORG  $2000
        MOVE.W #$4000,A0      STRING1 POINTER
        MOVE.W #$4100,A1      STRING2 POINTER
        MOVE.W #$4200,A2      STRING3 POINTER
        MOVEQ.L #7,D0         COUNT - 1
LOOP    MOVE.W (A0),D1
        EXT.L D1              EXTEND TO LGWRD
        DIVS #5,D1            FORM REMAINDER
        SWAP D1               PUT REMAINDER
        TST.W D1              IN LOW WORD
        BEQ.S EVENLY          EVENLY DIVISIBLE
        MOVE.W (A0)+,(A2)+    NO!
        BRA.S TEST
EVENLY  MOVE.W (A0)+,(A1)+    YES, IT IS!
TEST    DBRA D0,LOOP          LOOPED 8 TIMES?
        BRA.S *               YES, DONE!
        END
```

MT8-750-2

SOLUTION TO LAB DAY 4

```
*************************************
**    SOLUTION 2 (UNSIGNED NUMBERS)  **
*************************************

        ORG  $2000
        LEA  $4000,A0         STRING1 POINTER
        LEA  $4100,A1         STRING2 POINTER
        LEA  $4200,A2         STRING3 POINTER
        CLR.L D1
        MOVE.W #7,D0          COUNT - 1
LOOP    MOVE.W (A0),D1
        DIVU #5,D1            FORM REMAINDER
        LSR.L #8,D1           PUT REMAINDER
        LSR.L #8,D1           IN LOW WORD
        BEQ.S EVENLY          EVENLY DIVISIBLE?
        MOVE.W (A0)+,(A2)+    NO!
        BRA.S TEST
EVENLY  MOVE.W (A0)+,(A1)+    YES, IT IS!
TEST    DBRA D0,LOOP          LOOPED 8 TIMES?
        BRA.S *               YES, DONE!
        END
```

MT8-751-2

SOLUTION TO LAB DAY 4

```
                ************************************
**                SOLUTION 3  (UNSIGNED NUMBERS)                **
                ************************************

                ORG  $2000
                LEA  $4000,A0          STRING1  POINTER
                LEA  $4100,A1          STRING2  POINTER
                LEA  $4200,A2          STRING3 POINTER
                MOVE.W #7,D0           COUNT  - 1
LOOP            CLR.L  D1              UPPER WORD=0
                MOVE.W  (A0),D1
                DIVU #5,D1             FORM  REMAINDER
                ANDI.L  #$FFFF0000,D1  CHECK  REMAINDER
                BEQ.S EVENLY           EVENLY DIVISIBLE?
                MOVE.W  (A0)+,(A2)+    NO!
                BRA.S  TEST
EVENLY          MOVE.W (A0)+,(A1)+     YES,  IT  IS!
TEST            DBRA  D0,LOOP          LOOPED  8  TIMES?
                BRA.S  *               YES,  DONE!
                END
```

MT8-752-2