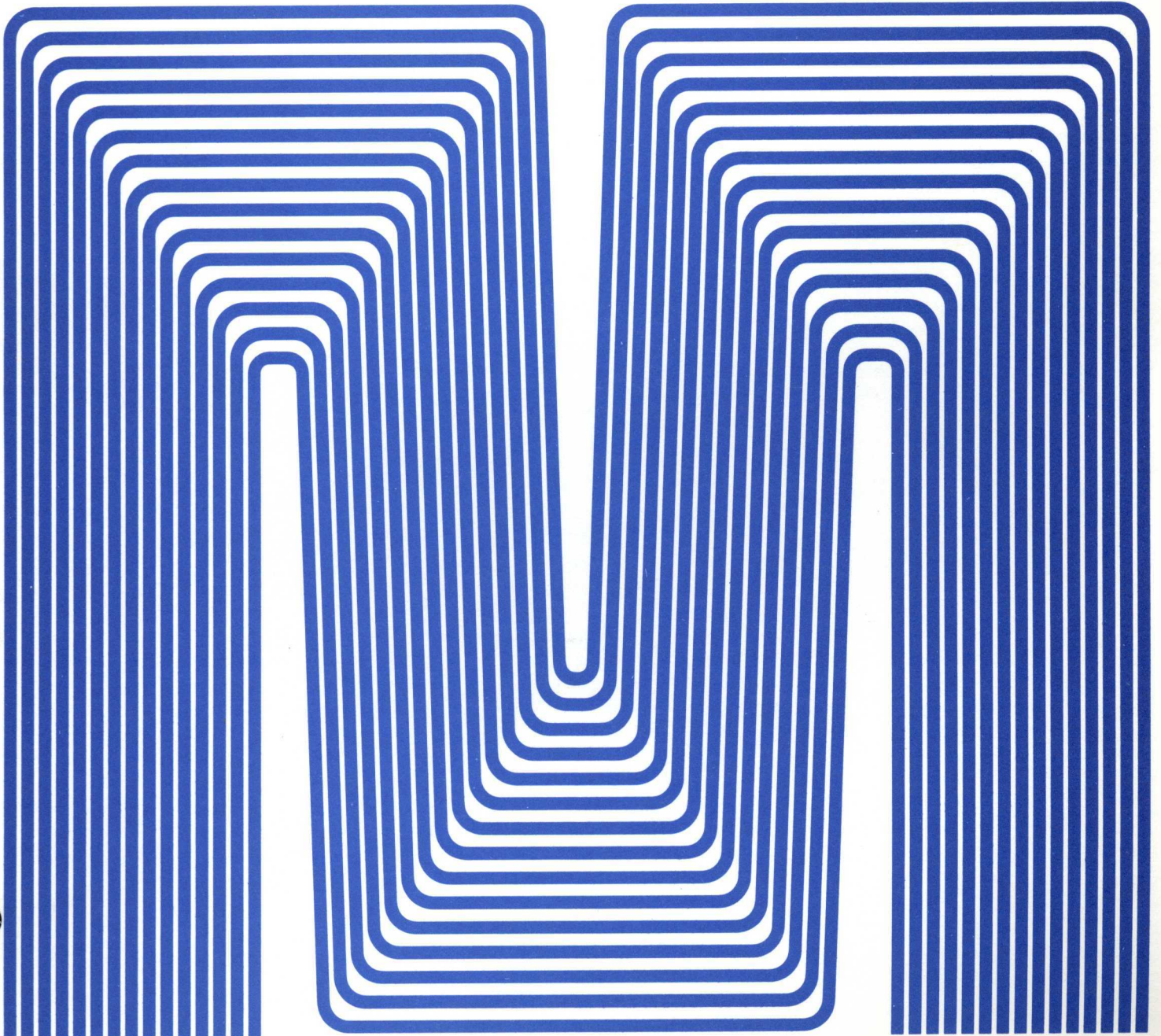


MicroSystems Inc.

FEB 20 1970

Micro 810 Computer

**AP810 Assembly
Program**

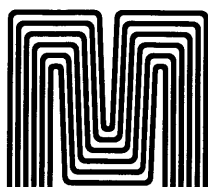


MICRO 810 COMPUTER

AP810 ASSEMBLY PROGRAM

69-2-0810-002

January 1970



Micro Systems Inc.

Micro Systems Inc
644 East Young Street
Santa Ana, California 92705

A Microdata Subsidiary

TABLE OF CONTENTS

1.	INTRODUCTION	1
2.	SOURCE LANGUAGE	3
	Statement Format	3
	Operand Field Expressions	4
3.	MACHINE INSTRUCTIONS	5
	Memory Addressing Modes	5
4.	ASSEMBLER INSTRUCTIONS	9
5.	ASSEMBLY LISTING	13
	Format	13
	Error Flags	13
	Sample Listing	14
6.	OBJECT PROGRAM CARD DECK	17

APPENDIXES

A.	Micro 810 Instructions	19
B.	Standard Character Codes	23
C.	Teletype Control and Transmission Codes	24
D.	Table of Powers of Two	25
E.	Hexadecimal – Decimal Integer Conversion Tables	26

ILLUSTRATIONS

1.	Sample Listing	15
----	--------------------------	----

1. INTRODUCTION

AP810 is a symbolic assembly program for the MICRO 810 computer. The assembler provides for symbolic memory addressing, and mnemonics for machine and assembler instructions. This program is written in FORTRAN IV and may be adapted to many computer systems. The MICRO 810 source program is entered by punch cards and the output of the assembler includes an assembly listing and an object program card deck. This assembler generates non-relocatable programs.

The AP810 assembly language includes the following features:

- Address Arithmetic — Decimal and hexadecimal numbers, symbolic addresses, and arithmetic expressions.
- Data definitions — Variable precision integer, address expressions, character constants, and literals may be designated.
- Listing control — The format of the listing may be controlled and comment cards included.
- Diagnostics — Diagnostics for source program errors are included in the output listing.

2. SOURCE LANGUAGE

The source language is a sequence of symbolic instructions, called statements which are punched on cards. Each statement is punched on a single card. Each statement may consist of from one to four entries: a name field, an operation field, an operand field, and a comment field. Columns 73-80 are normally used for identification or sequence numbers. Entries in the operand field are expressions which may consist of decimal numbers, hexadecimal numbers and symbolic values.

STATEMENT FORMAT

Name Field

The name field entry is a symbol composed of from one to six characters starting with column 1 and terminating with the first blank. The first character of a symbol is alphabetic or period, subsequent characters may be alphabetic, numeric or a period. A name entry is usually optional and the type of instruction determines the legal content of the name field. The symbol takes on the current value of the assembler's location counter unless assigned another value by an assembler instruction. When an asterisk (*) appears in column 1 the remainder of the line is considered as comment and is not processed by the assembler except to place it on the listing.

Operation Field

The operation field entry is a mnemonic operation code specifying the machine or assembler instruction. The field begins in column 8 and is terminated by the first blank. Certain memory referencing instruction modes use special symbols suffixed to the mnemonic.

Operand Field

The operand field entries identify and describe data to be acted upon by instructions as, for example, memory locations, or literals. One or more operands may be written, depending on the needs of the instruction. Entries are separated by commas, and no blanks may appear in the field. The operand field may start with any column following the operation field, but must start by column 40. Column 14 is the normal starting column. It is terminated by the first blank column.

Comments Field

Comments describing the information about the program may be inserted between the end of the operand field and column 72. All characters, including spaces, may be used in writing a comment.

OPERAND FIELD EXPRESSIONS

Expressions in the operand field are made up of one or more terms which are connected by + and – arithmetic operators. No parenthetical expressions are allowed. Each term of the expression represents a value. Values may be assigned by the assembler program (symbols), or there may be inherent in the term itself, (constants). The range of values depends on the operand and the instruction. Address expressions for relative type addressing are written as if they are not relative. The assembler will convert these expressions to a relative displacement.

Symbols

A symbol is composed of one to six characters. The first character must be alphabetic or period, subsequent characters may be numeric, alphabetic or period. Imbedded blanks are not allowed and the assembler stops scanning the symbol with the first character which is not alphanumeric or a period. All symbols, except the special symbols * and **, used in an operand field must be defined by a single appearance in the name field of statement within the program.

Special Symbols

The special symbol, *, represents the momentary value of the assemblers location counter. It may be used as any other symbol in an expression but must never appear in the name field. When used in the operand field of a multi-byte instruction it will assume the value of the address of the first byte of the instruction.

The special symbol, **, may be used to indicate that the field will be changed during program execution. Normally it is used by itself without any other terms in the expression. The symbol yields a zero value and the byte length is governed by the usage.

Constants

The values of the constant terms are not assigned by the assembler program, but are inherent in the terms. There are two types of constant terms: decimal and hexadecimal.

a. Decimal Constant

A decimal constant is an unsigned decimal number. The value must be less than 65,536.

b. Hexadecimal Constant

A hexadecimal constant is an unsigned hexadecimal number of up to four characters written as a sequence of hexadecimal digits. The digits are enclosed in single quotation marks and preceded by the letter X. Each hexadecimal digit represents a four-bit binary number. The characters A through F are used to identify the hexadecimal integers 10 through 15.

3. MACHINE INSTRUCTIONS

Machine language instructions occupy one to five bytes of memory. The first byte contains the operation code and addressing mode code. The instruction is expressed in the assembly language by a three character mnemonic code in the operation field. Memory referencing instructions have eight addressing modes which occupy eight consecutive instruction codes. The addressing mode is expressed by the value of the address expression in the operand field and by a special symbol suffixed to the mnemonic operation code.

If the instruction is a memory referencing literal (mode 7) type, the second through fifth bytes of the instruction hold a literal which is expressed as a constant in the operand field. The types of constants and the way in which they are assembled is described later in this section under Mode 7. In general the literals are written and assembled in a manner similar to the DC assembler instruction.

Non-literal memory referencing instructions require 0, 1, or 2 bytes following the first byte to hold an address. Modes which make use of one byte for address contain an eight bit address of the first 256 memory locations (page 0), an eight bit signed displacement, or an 8 bit unsigned bias. Modes which make use of two bytes for the address hold a 16-bit address word with an index flag in bit 15.

The shift instructions use the second byte of the instruction to hold an eight bit shift count. The conditional jump instructions use the second byte to hold an eight bit signed displacement.

Examples of the method of writing machine instructions in the assembly language are shown in the sample listing at the end of section 5.

MEMORY ADDRESSING MODES

Each memory referencing instruction has eight addressing modes. The operation of the computer in each of these modes is explained in the MICRO 810 Computer Reference Manual. All addresses for modes 0-3 are written without regard for the mode to be used. The assembler will convert to relative addresses if the evaluated address is not in page 0.

Mode 0: Direct Page 0

When the evaluated address in the operand field is less than 256 and the mnemonic operation code is suffixed with a blank, the instruction is assembled as mode 0, and an eight bit address is placed in the second byte of the instruction.

Mode 1: Direct Relative

When the evaluated address in the operand field is greater than or equal to 256 and the mnemonic operation code is suffixed with a blank, the instruction is assembled as mode 1. If the address minus the location of the first byte of the instruction minus two is in the range +127 to -128 this value is assembled into the second byte of the instruction; otherwise a zero value is assembled and an 'R' diagnostic flag is placed in the listing.

Mode 2: Indirect Page 0

When the evaluated address in the operand field is less than 256 and the mnemonic operation code is suffixed with an *, the instruction is assembled as mode 2, and an eight bit address is placed in the second byte of the instruction.

Mode 3: Indirect Relative

When the evaluated address in the operand field is greater than or equal to 256 and the mnemonic operation code is suffixed with an *, the instruction is assembled as mode 3. If the address minus the location of the first byte of the instruction minus two is in the range +127 to -128 the value is assembled into the second byte of the instruction; otherwise a zero value is assembled and an 'R' diagnostic flag is placed in the listing.

Mode 4: Indexed

When the instruction mnemonic operation code is suffixed with -, the instruction is assembled as mode 4 in a single byte instruction. The operand field is disregarded.

Mode 5: Indexed With Bias

When the instruction mnemonic operation code is suffixed with a +, the instruction is assembled as mode 5. If the value of the operand is less than 256 it is assembled into the second byte of the instruction; otherwise a zero value is assembled and an 'A' diagnostic flag is placed in the listing.

Mode 6: Extended Address

When the instruction mnemonic operation code is suffixed with a /, the instruction is assembled as mode 6. The value of the first operand is assembled as a 16-bit address in the second and third bytes of the instruction. If the character X preceded by a comma, is entered as a second operand the index flag in bit 15 of the address word is set to a 1-bit.

Mode 7: Indirect Extended Address (Jump and Return Jump)

When the instruction mnemonic operation code is suffixed with an =, the instruction is assembled as mode 7. The value of the first operand is assembled as a 16-bit address in the second and third bytes of the instruction. If the character X, preceded by a comma, is entered as a second operand the index flag in bit 15 of the address word is set to a 1-bit.

Mode 7: Literal

When the instruction mnemonic operation code is suffixed with an =, the instruction is assembled as mode 7 and the value of the operand field is assembled into the 1-4 bytes following the first byte. The length of a literal for the variable word length operand instructions (third character of mnemonic is V) depends on the literal type or the number of digits or characters. The programmer must assure that the length of these operands is consistent with the word length mode in effect. Instruction which are fixed word length operand type always assemble a two byte literal. A literal is enclosed in single quotation marks and is preceded by a single type identifier character. A literal without a type identifier and quotes is evaluated as a 16-bit address type.

H – Half Word Integer Literal

A half word integer constant is written as a signed or unsigned decimal number in the range +127 to -128. The number is converted to its binary equivalent and assembled into a single byte if a variable operand length instruction otherwise it is assembled into a two byte operand and right justified with leading zeros.

F – Full Word Integer Literal

A full word integer constant is written as a signed or unsigned decimal number in the range $+2^{15}-1$ to -2^{15} . The number is converted to its binary equivalent and assembled in two eight bit bytes.

E – Extended Word Integer Literal

An extended word integer constant is written as a signed or unsigned decimal number in the range $+2^{23}-1$ to -2^{24} . The number is converted to its binary equivalent and assembled in three eight bit bytes if the instruction is a variable operand length type; otherwise two bytes of zero are assembled and an 'A' diagnostic is placed in the listing.

D – Double Word Integer Literal

A double word integer constant is written as a signed or unsigned decimal number in the range $+2^{31}-1$ to -2^{31} . The number is converted to its binary equivalent and assembled in four eight bit bytes if the instruction is a variable length type; otherwise two bytes of zero are assembled and an 'A' diagnostic is placed in the listing.

A – Address Constant

A 16-bit address constant is written as an expression consisting of decimal numbers, hexadecimal numbers and symbols. The value of the expression must be in the range 0 to 65535 and any symbol which is undefined will result in a diagnostic error flag and a zero value. If the location counter symbol * appears in the address expression it will assume the value of the address of the first byte of the instruction. The index flag in bit 15 of the address word is set to a 1-bit by following the expression with a second operand consisting of the character X. The two operands are separated by a comma. An address constant may also be written without the A type identifier and quotes.

X – Hexadecimal Literal

A hexadecimal constant consists of one to eight hexadecimal digits (0-9 and A-F). If the instruction is a variable word length operand type, the number of bytes assembled is determined by the number of digits. When the number of digits is odd a leading zero is added. If the instruction is a fixed word length type, the constant is right justified with leading zeros in two eight-bit bytes, and when the number of digits is greater than four, an 'A' diagnostic is placed in the listing.

C – Character Literal

A character constant consists of a sequence of one to four characters, excluding the single quotation mark. Each character is converted to its ASCII value which is assembled as a single eight bit byte. If the instruction is a fixed word length type an 'A' diagnostic and zero value constant occurs if the number of characters is greater than two; if only a single character is included it is assembled right justified with leading zeros in two eight-bit bytes.

4. ASSEMBLER INSTRUCTIONS

Seven assembler instructions are included for control of the assembly process and the output listing.

ORG – Set Location Counter

The ORG assembler instruction alters the setting of the location counter. The name field entry, if any, will be assigned the value of the program counter after it is altered. The operand field of ORG must contain an expression whose value will be placed in the location counter. All symbols in the expression must have been previously defined when the instruction is first encountered. The next instruction which places object code in the program is forced to begin a new object card.

EQU – Equate Symbol

The EQU assembler instruction is used to define a symbol by assigning to it the value of the operand field. Any symbols appearing in the expression must have been previously defined when the instruction is first encountered. A name field entry must be present.

SET – Set Symbol

The SET assembler instruction assigned the value of the operand field to the symbol in the name field. Any symbols appearing in the operand field must have been previously defined when the instruction is first encountered. A name field entry must be present and may be a symbol previously defined. A multi-defined error cannot occur.

DC – Define Constant

The DC assembler instruction is used to provide constant data in memory. Each statement specifies only one constant. The constants which may be specified are:

8-bit decimal, 16-bit decimal, 24-bit decimal, 32-bit decimal, address, hexadecimal, and character.

A constant is enclosed in single quotation marks and is preceded by a single type identifier character. A constant without a type identifier and quotes is evaluated as a 16-bit address type. The method of expressing constants for the DC instruction is the same as for literals with the variable word length memory referencing instructions. The seven constant identifiers and a description of each follows:

H – Half Word Integer Constant

A half word integer constant is written as a signed or unsigned decimal number in the range +127 to –128. The number is converted to its binary equivalent and assembled as a single eight-bit byte.

F – Full Word Integer Constant

A full word integer constant is written as a signed or unsigned decimal number in the range $+2^{15}-1$ to -2^{15} . The number is converted to its binary equivalent and assembled in two eight-bit bytes.

E – Extended Word Integer Constant

An extended word integer constant is written as a signed or unsigned decimal number in the range $+2^{23}-1$ to -2^{24} . The number is converted to its binary equivalent and assembled in three eight-bit bytes.

D – Double Word Integer Constant

A double word integer constant is written as a signed or unsigned decimal number in the range $+2^{31}-1$ to -2^{31} . The number is converted to its binary equivalent and assembled in four eight-bit bytes.

A – Address Constant

A 16-bit address constant is written as an expression consisting of decimal numbers, hexadecimal numbers and symbols. The value of the expression must be in the range 0 to 65535 and any symbol which is undefined will result in a diagnostic error flag and a zero value. If the location counter symbol * appears in the address expression it will assume the value of the address of the first byte of the instruction. The index flag in bit 15 of the address word is set to a 1-bit by following the expression by a second operand consisting of the character X. The two operands are separated by a comma. An address constant may also be written without the A type identifier and quotes.

X – Hexadecimal Constant

A hexadecimal constant consists of one to eight hexadecimal digits, (0-9 and A-F). The number of bytes assembled is determined by the number of digits. If the number of digits is odd a leading zero is added.

C – Character Constant

A character constant consists of a sequence of characters, excluding the single quotation mark. The constant must be terminated by column 72. Each character is converted to its ASCII value and assembled as a single eight bit byte

DS – Define Storage

The DS instruction is used to reserve areas of memory and to assign names to these areas. The value of the operand field specifies the number of bytes to be reserved. A zero value in the operand field caused no bytes to be reserved, but causes the symbol in the name field to take on the current value of the location counter. No object code is assembled into the binary program and the next instruction which does place code in the program is forced to begin a new object card.

IDENT – **Program Identification**

The IDENT assembler instruction is used to identify the start of a program and to supply the program name which is located in the operand field. The IDENT must be the first statement in a source program.

END – **End Assembly**

The END assembler instruction terminates the assembly of a program and must be the last statement in a source program.

SPACE – **Space Listing**

The SPACE assembler instruction causes one or more blank lines to be inserted into the listing. The name field is disregarded by the assembler. The operand field contains an expression specifying the number of blank lines. If the spacing is beyond the end of the current page, the listing begins at the top of the next page.

EJECT – **Start New Listing Page**

The EJECT instruction causes the next line of the listing to appear at the top of the next page. The name and operand fields are disregarded by the assembler.

5. ASSEMBLY LISTING

The output listing from AP810 contains the memory address, and contents of words in the object program. The source statement is printed side-by-side with the object code.

FORMAT

Printer Columns	Contents
5 – 8	Error flags
11 – 14	Memory location of first byte of instruction, constant or reserved memory. Memory addresses are not printed for SET and EQU assembler instructions.
17 – 18	Instruction operation code. This field is blank for assembler instructions.
19 – 26	Operand
32 – 111	Source statement

ERROR FLAGS

A Address Error

This error occurs when an address expression in the operand field is incorrectly written or the value is out of range for one of the operands. An error flag will occur for each operand in error or out of range.

M Multidefined Symbol Error

This error occurs when the symbol in the name field has been previously defined by appearing in the name field of another instruction.

N Name Field Error

This error flag occurs when the symbol in the name field starts with a character other than alphabetic or period, or contains a non alphanumeric or non period character.

O Operation Mnemonic Error

This error occurs when the assembler does not recognize the contents of the operation field starting in column 8. A two byte zero value is assembled to allow patching.

R Range Error

This error occurs when the operand address of a relative addressing mode memory reference instruction or a Conditional Jump instruction is not in the range of +127 to -128 bytes from the location of the next instruction.

U Undefined Symbol Error

This error occurs when the symbol encountered in an expression of the operand field is not defined by an appearance in the name field.

SAMPLE LISTING

The sample listing on the next page shows the format of the listing and provides examples of how to write each instruction type, literals, constants, and assembler instructions. The six types of error conditions are also illustrated.

```

IDENT SAMPLE
* THIS SAMPLE PROGRAM SHOWS HOW TO WRITE EACH INSTRUCTION TYPE

** CONTROL INSTRUCTIONS
0000 00 START HLT NO OPERAND FIELD FOR CONTROL INSTRUCTIONS
0001 34 NOP

** CONDITIONAL JUMP INSTRUCTIONS
0002 1000 CJ JOV **2 A ZERO DISPLACEMENT ADDRESSES NEXT INST
0004 11FC JAZ 2 A NEGATIVE DISPLACEMENT
0006 1EFA NAB CJ OPERAND FIELD STARTS AFTER MNEMONIC
R 0008 1F00 MAX **300 THIS OPERAND ADDRESS IS OUT OF RANGE
O 000A 0000 SSS CJ+2 ILLEGAL MNEMONIC OPERATION CODE

** SHIFT INSTRUCTIONS
000C 2005 ALPHA LLA 5
A 000E 2200 LLL -5 NEGATIVE OPERANDS ARE NOT ALLOWED FOR SHIFT
0010 2810 ALA X'10' SHIFT COUNT MAY BE EXPRESSION
M 0012 2F0C ALPHA ARL TEN+2 SYMBOL IN NAME IS MULTI-DEFINED
0014 2800 ALA ** ** INDICATES TO BE CHANGED DURING EXECUT.

** INPUT/OUTPUT
0016 3000 BETA IHS NO OPERAND NEEDED FOR SERIAL I/O
0018 3142 IBA 2,2 DEVICE ORDER, DEVICE NUMBER
001A 3B848146 ORM 4,4,BUF,X MEMORY I/O WITH INDEX FLAG - X

** REGISTER OPERATE
001E 41 XKA NO OPERAND FOR REGISTER INSTRUCTIONS

** NON LITERAL MEMORY REFERENCING INSTRUCTIONS
N 001F E01F 123456 LDA * MODE 0 SYMBOL IN NAME IS NOT LEGAL
0100 F100 .DOT STA **2 MODE 1
U 0102 8200 LDX* 00PS MODE 2 SYMBOL IN OPERAND IS UNDEFINED
0104 8345 LDX* BUF+5 MODE 3
0106 8C A23456 STX- MODE 4 NO OPERAND FIELD FOR THIS MODE
0107 A01E ADV+ 30 MODE 5
0109 96813A MUL/ SAM,X MODE 6 WITH ADDRESS FLAG
010C EF013A LDV/ SAM MODE 6 NO INDEX FLAG

** FIXED WORD LENGTH LITERAL MODE INSTRUCTIONS (MODE 7)
010F E700FF LDA= H'-1' TWO BYTE LITERAL WITH FIRST BYTE ZERO
A 0112 F70000 SPOT STA= ** A REG. STORED IN SPOT+1
0115 970000 MBL= E'100' E AND D TYPES ARE NOT LEGAL IN FIXED WORD
0118 E7000C LDA= A'ALPHA' ADDRESS TYPE WITH NO INDEX FLAG
011B E7800C LDA= A'ALPHA,X' ADDRESS TYPE WITH INDEX FLAG - X
011E 87000C LDX= ALPHA ADDRESS TYPE LITERAL WITHOUT A'
0121 C700C1 CPA= C'A' TWO BYTE LITERAL WITH FIRST BYTE ZEROS
0124 0700FF ANA= X'FF' TWO BYTE LITERAL WITH HEX RIGHT JUSTIFIED

** VARIABLE WORD LENGTH LITERALS (MODE 7)
0127 EF013A LDV= SAM TWO BYTE ADDRESS LITERAL
012A AF000003FF ADV= D'1023' FOUR BYTE INTEGER
012F DF09ABCDEF ANV= X'9ABCDEF' FOUR BYTE, LENGTH DETERMINED BY COUNT
0134 CFC1 CPV= C'A' SINGLE BYTE LITERAL, LENGTH DET. BY COUNT

** ASSEMBLER INSTRUCTIONS
0008 TEN EQU 8 SYMBOL IS EQUATED TO 8
000A TEN SFT 10 SET DOES NOT ALLOW MULTI-DEFINED ERROR
0136 800B DC A'TEN+2-1,X' ADDRESS CONSTANT
0138 800B DC TEN+2-1,X ADDRESS CONSTANT
013A 04C8C9D3 SAM DC C'THIS IS A DC'
0146 BUF DS 4 FOUR BYTES ARE RESERVED
014A 0000 DC ** WORD TO BE CHANGED DURING EXECUTION

** JUMP AND RETURN JUMP INSTRUCTIONS
014C 6000C JMP/ ALPHA MODE 6 TWO BYTE EXTENDED ADDRESS
014F 67800C JMP= ALPHA,X MODE 7 TWO BYTE INDIRECT EXTENDED ADDRESS
0152 000C END ALPHA OPERAND ADDRESS IS EXECUTE ADDRESS

```

FIGURE 1. SAMPLE LISTING

6. OBJECT PROGRAM CARD DECK

The AP810 assembly program generates a deck of cards which contain the binary object code. All information punched on the cards is in Hollerith code, with a single hexadecimal digit (four binary bits) punch in each column. This format allows easy visual reading of the cards after they are interpreted and permits rapid patching or generation of patches to the deck.

The cards have two fields as follows:

- | | |
|--------------|-------------------------------------|
| Column 1-4 | – Load or execution address |
| Columns 5-80 | – Object code, two columns per byte |

The card is terminated with the first blank column. If the card is blank except for an address in columns 1-4 the address is an execution address where the loader will transfer when the card is encountered. DS and ORG assembler instructions cause new cards to be started since a new load address is required. No information is loaded for the memory locations reserved by a DS instruction.

APPENDIXES

APPENDIX A

MICRO 810 INSTRUCTIONS

Operation Code	Mnemonic	Operand	Instruction Name
Control			
00	HLT		Halt
01	TRP		Trap
02	ESW		Enter Sense Switches
03	PMP		Protect Memory Page
04	DIN		Disable Interrupt System
05	EIN		Enable Interrupt System
06	DRT		Disable Real Time Clock
07	ERT		Enable Real Time Clock
08	RO1		Reset Overflow and Set Word Length to 1
09	RO2		Reset Overflow and Set Word Length to 2
0A	RO3		Reset Overflow and Set Word Length to 3
0B	RO4		Reset Overflow and Set Word Length to 4
0C	SO1		Set Overflow and Set Word Length to 1
0D	SO2		Set Overflow and Set Word Length to 2
0E	SO3		Set Overflow and Set Word Length to 3
0F	SO4		Set Overflow and Set Word Length to 4
34	NOP		No Operation
Conditional Jump			
10	JOV	a	Jump if Overflow Set
11	JAZ	a	Jump if A Equal to Zero
12	JBZ	a	Jump if B Equal to Zero
13	JXZ	a	Jump if X Equal to Zero
14	JAN	a	Jump if A Negative
15	JXN	a	Jump if X Negative
16	JAB	a	Jump if A Equals B
17	JAX	a	Jump if A Equals X
18	NOV	a	Jump if Overflow Not Set
19	NAZ	a	Jump if A Not Equal to Zero
1A	NBZ	a	Jump if B Not Equal to Zero
1B	NXZ	a	Jump if X Not Equal to Zero
1C	NAN	a	Jump if A Not Negative
1D	NXN	a	Jump if X Not Negative
1E	NAB	a	Jump if A Not Equal to B
1F	NAX	a	Jump if A Not Equal to X

Operation Code	Mnemonic	Operand	Instruction Name
----------------	----------	---------	------------------

Shift

20	LLA	n	Logical Left A
21	LLB	n	Logical Left B
22	LLL	n	Logical Left Long
24	LRA	n	Logical Right A
25	LRB	n	Logical Right B
26	LRL	n	Logical Right Long
28	ALA	n	Arithmetic Left A
29	ALB	n	Arithmetic Left B
2A	ALL	n	Arithmetic Left Long
2C	ARA	n	Arithmetic Right A
2D	ARB	n	Arithmetic Right B
2E	ARL	n	Arithmetic Right Long

Input/Output

30	IBS		Input Byte Serially
31	IBA	f,d	Input Byte to A
32	IBB	f,d	Input Byte to B
33	IBM	f,d,a or a,X	Input Byte to Memory
38	OBS		Output Byte Serially
39	OBA	f,d	Output Byte from A
3A	OBB	f,d	Output Byte from B
3B	OBM	f,d,a or a,X	Output Byte from Memory

Register Operate

40	ORA		OR B with A
41	XRA		Exclusive – OR B with A
42	ORB		OR A with B
43	XRB		Exclusive – OR A with B
44	INX		Increment X
45	DCX		Decrement X
46	AWX		Add Word Length to X
47	SWX		Subtract Word Length from X
48	INA		Increment A
49	INB		Increment B
4A	OCA		One's Complement A
4B	OCB		One's Complement B
4C	TAX		Transfer A to X
4D	TBX		Transfer B to X
4E	TXA		Transfer X to A
4F	TXB		Transfer X to B

Operation Code	Mnemonic	Operand	Instruction Name
Memory Reference			
60	JMP	a or a,X or l	Jump
68	RTJ	a or a,X or l	Return Jump
70	IWM	a or a,X or l	Increment Word in Memory
78	DWM	a or a,X or l	Decrement Word in Memory
80	LDX	a or a,X or l	Load X
88	STX	a or a,X or l	Store X
90	MUL	a or a,X or l	Multiply
98	DIV	a or a,X or l	Divide
A0	ADA	a or a,X or l	Add to A
A8	ADV	a or a,X or l	Add Variable
B0	SBA	a or a,X or l	Subtract from A
B8	SBV	a or a,X or l	Subtract Variable
C0	CPA	a or a,X or l	Compare A
C8	CPV	a or a,X or l	Compare Variable
D0	ANA	a or a,X or l	And A
D8	ANV	a or a,X or l	And Variable
E0	LDA	a or a,X or l	Load A
E8	LDV	a or a,X or l	Load Variable
F0	STA	a or a,X or l	Store A
F8	STV	a or a,X or l	Store Variable

Operation Code	Mnemonic	Operand	Instruction Name and Mode
Jump and Return Jump Addressing Modes (typical)			
60	JMP	a	Jump — Page 0
61	JMP	a	Jump — Relative
62	JMP*	a	Jump — Indirect Page 0
63	JMP*	a	Jump — Indirect Relative
64	JMP—		Jump — Indexed
65	JMP+	a	Jump — Indexed with Bias
66	JMP/	a or a,X	Jump — Extended Address
67	JMP=	a or a,X	Jump — Indirect Extended Address

Operation Code	Mnemonic	Operand	Instruction Name and Mode
----------------	----------	---------	---------------------------

Other Memory Referencing Addressing Modes (typical)

A0	ADA	a	Add to A – Page 0
A1	ADA	a	Add to A – Relative
A2	ADA*	a	Add to A – Indirect Page 0
A3	ADA*	a	Add to A – Indirect Relative
A4	ADA-		Add to A – Indexed
A5	ADA+	a	Add to A – Indexed With Bias
A6	ADA/	a	Add to A – Extended Address
A7	ADA=	l	Add to A – Literal

Legend of Operand Symbols

a	– Address expression
a,X	– Address expression with index flag
l	– Literal
f	– Device Order
d	– Device Number
n	– Shift Count

APPENDIX B

STANDARD CHARACTER CODES

SYMBOL ASCII EBCDIC HOLLERITH				SYMBOL ASCII EBCDIC HOLLERITH					
	(HEX)	(HEX)	(029)	(026)		(HEX)	(HEX)	(029)	(026)
blank	A0	40		blank	@	C0	7C	8-4	0-8-2
!	A1	5A		11-8-2	A	C1	C1		12-1
"	A2	7F	8-7	0-8-5	B	C2	C2		12-2
#	A3	7B	8-3	0-8-7	C	C3	C3		12-3
\$	A4	5B		11-8-3	D	C4	C4		12-4
%	A5	6C	0-8-4	11-8-7	E	C5	C5		12-5
&	A6	50	12	12-8-7	F	C6	C6		12-6
'	A7	7D	8-5	8-4	G	C7	C7		12-7
(A8	4D	12-8-5	0-8-4	H	C8	C8		12-8
)	A9	5D	11-8-5	12-8-4	I	C9	C9		12-9
*	AA	5C		11-8-4	J	CA	D1		11-1
+	AB	4E	12-8-6	12	K	CB	D2		11-2
,	AC	6B		0-8-3	L	CC	D3		11-3
-	AD	60		11	M	CD	D4		11-4
.	AE	4B		12-8-3	N	CE	D5		11-5
/	AF	61		0-1	O	CF	D6		11-6
0	B0	F0		0	P	D0	D7		11-7
1	B1	F1		1	Q	D1	D8		11-8
2	B2	F2		2	R	D2	D9		11-9
3	B3	F3		3	S	D3	E2		0-2
4	B4	F4		4	T	D4	E3		0-3
5	B5	F5		5	U	D5	E4		0-4
6	B6	F6		6	V	D6	E5		0-5
7	B7	F7		7	W	D7	E6		0-6
8	B8	F8		8	X	D8	E7		0-7
9	B9	F9		9	Y	D9	E8		0-8
:	BA	7A	8-2	8-5	Z	DA	E9		0-9
;	BB	5E		11-8-6	[DB	4F	12-8-7	12-8-5
<	BC	4C	12-8-4	12-8-6	\	DC	4A	12-8-2	0-8-6
=	BD	7E	8-6	8-3]	DD	5F	11-8-7	11-8-5
>	BE	6E	0-8-6	8-6	↑	DE	6D	0-8-5	8-7
?	BF	6F	0-8-7	12-8-2	←	DF	6A	0-8-2	8-2

APPENDIX C

TELETYPE CONTROL AND TRANSMISSION CODES

FUNCTION	ASCII
NULL	80
SOM (Print on)	81
EAO	82
EOM	83
EOT (Print off)	84
WRU	85
RU	86
BELL	87
FEO	88
H.TAB	89
LINE FEED	8A
V.TAB	8B
FORM	8C
CARRIAGE RETURN	8D
SO	8E
SI	8F
DCO	90
X-ON (Reader on)	91
TAPE (Punch on)	92
X-OFF (Reader off)	93
TAPE OFF (Punch off)	94
ERROR	95
SYNC	96
LEM	97
S0	98
S1	99
S2	9A
S3	9B
S4	9C
S5	9D
S6	9E
S7	9F

APPENDIX D

TABLE OF POWERS OF TWO

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25

APPENDIX E

HEXADECIMAL – DECIMAL INTEGER CONVERSION TABLES

The following tables aid in converting hexadecimal values to decimal values, or the reverse.

Direct Conversion Table

This table provides direct conversion of decimal and hexadecimal numbers in these ranges:

HEXADECIMAL	DECIMAL
000 to FFF	0000 to 4095

For numbers outside the range of the table, add the following values to the table figures:

HEXADECIMAL	DECIMAL
1000	4096
2000	8192
3000	12288
4000	16384
5000	20480
6000	24576
7000	28672
8000	32768
9000	36864
A000	40960
B000	45056
C000	49152
D000	53248
E000	57344
F000	61440

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00_	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010	0011	0012	0013	0014	0015
01_	0016	0017	0018	0019	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	0030	0031
02_	0032	0033	0034	0035	0036	0037	0038	0039	0040	0041	0042	0043	0044	0045	0046	0047
03_	0048	0049	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	0060	0061	0062	0063
04_	0064	0065	0066	0067	0068	0069	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079
05_	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	0090	0091	0092	0093	0094	0095
06_	0096	0097	0098	0099	0100	0101	0102	0103	0104	0105	0106	0107	0108	0109	0110	0111
07_	0112	0113	0114	0115	0116	0117	0118	0119	0120	0121	0122	0123	0124	0125	0126	0127
08_	0128	0129	0130	0131	0132	0133	0134	0135	0136	0137	0138	0139	0140	0141	0142	0143
09_	0144	0145	0146	0147	0148	0149	0150	0151	0152	0153	0154	0155	0156	0157	0158	0159
0A_	0160	0161	0162	0163	0164	0165	0166	0167	0168	0169	0170	0171	0172	0173	0174	0175
0B_	0176	0177	0178	0179	0180	0181	0182	0183	0184	0185	0186	0187	0188	0189	0190	0191
0C_	0192	0193	0194	0195	0196	0197	0198	0199	0200	0201	0202	0203	0204	0205	0206	0207
0D_	0208	0209	0210	0211	0212	0213	0214	0215	0216	0217	0218	0219	0220	0221	0222	0223
0E_	0224	0225	0226	0227	0228	0229	0230	0231	0232	0233	0234	0235	0236	0237	0238	0239
0F_	0240	0241	0242	0243	0244	0245	0246	0247	0248	0249	0250	0251	0252	0253	0254	0255
10_	0256	0257	0258	0259	0260	0261	0262	0263	0264	0265	0266	0267	0268	0269	0270	0271
11_	0272	0273	0274	0275	0276	0277	0278	0279	0280	0281	0282	0283	0284	0285	0286	0287
12_	0288	0289	0290	0291	0292	0293	0294	0295	0296	0297	0298	0299	0300	0301	0302	0303
13_	0304	0305	0306	0307	0308	0309	0310	0311	0312	0313	0314	0315	0316	0317	0318	0319
14_	0320	0321	0322	0323	0324	0325	0326	0327	0328	0329	0330	0331	0332	0333	0334	0335
15_	0336	0337	0338	0339	0340	0341	0342	0343	0344	0345	0346	0347	0348	0349	0350	0351
16_	0352	0353	0354	0355	0356	0357	0358	0359	0360	0361	0362	0363	0364	0365	0366	0367
17_	0368	0369	0370	0371	0372	0373	0374	0375	0376	0377	0378	0379	0380	0381	0382	0383
18_	0384	0385	0386	0387	0388	0389	0390	0391	0392	0393	0394	0395	0396	0397	0398	0399
19_	0400	0401	0402	0403	0404	0405	0406	0407	0408	0409	0410	0411	0412	0413	0414	0415
1A_	0416	0417	0418	0419	0420	0421	0422	0423	0424	0425	0426	0427	0428	0429	0430	0431
1B_	0432	0433	0434	0435	0436	0437	0438	0439	0440	0441	0442	0443	0444	0445	0446	0447
1C_	0448	0449	0450	0451	0452	0453	0454	0455	0456	0457	0458	0459	0460	0461	0462	0463
1D_	0464	0465	0466	0467	0468	0469	0470	0471	0472	0473	0474	0475	0476	0477	0478	0479
1E_	0480	0481	0482	0483	0484	0485	0486	0487	0488	0489	0490	0491	0492	0493	0494	0495
1F_	0496	0497	0498	0499	0500	0501	0502	0503	0504	0505	0506	0507	0508	0509	0510	0511

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20_	0512	0513	0514	0515	0516	0517	0518	0519	0520	0521	0522	0523	0524	0525	0526	0527
21_	0528	0529	0530	0531	0532	0533	0534	0535	0536	0537	0538	0539	0540	0541	0542	0543
22_	0544	0545	0546	0547	0548	0549	0550	0551	0552	0553	0554	0555	0556	0557	0558	0559
23_	0560	0561	0562	0563	0564	0565	0566	0567	0568	0569	0570	0571	0572	0573	0574	0575
24_	0576	0577	0578	0579	0580	0581	0582	0583	0584	0585	0586	0587	0588	0589	0590	0591
25_	0592	0593	0594	0595	0596	0597	0598	0599	0600	0601	0602	0603	0604	0605	0606	0607
26_	0608	0609	0610	0611	0612	0613	0614	0615	0616	0617	0618	0619	0620	0621	0622	0623
27_	0624	0625	0626	0627	0628	0629	0630	0631	0632	0633	0634	0635	0636	0637	0638	0639
28_	0640	0641	0642	0643	0644	0645	0646	0647	0648	0649	0650	0651	0652	0653	0654	0655
29_	0656	0657	0658	0659	0660	0661	0662	0663	0664	0665	0666	0667	0668	0669	0670	0671
2A_	0672	0673	0674	0675	0676	0677	0678	0679	0680	0681	0682	0683	0684	0685	0686	0687
2B_	0688	0689	0690	0691	0692	0693	0694	0695	0696	0697	0698	0699	0700	0701	0702	0703
2C_	0704	0705	0706	0707	0708	0709	0710	0711	0712	0713	0714	0715	0716	0717	0718	0719
2D_	0720	0721	0722	0723	0724	0725	0726	0727	0728	0729	0730	0731	0732	0733	0734	0735
2E_	0736	0737	0738	0739	0740	0741	0742	0743	0744	0745	0746	0747	0748	0749	0750	0751
2F_	0752	0753	0754	0755	0756	0757	0758	0759	0760	0761	0762	0763	0764	0765	0766	0767
30_	0768	0769	0770	0771	0772	0773	0774	0775	0776	0777	0778	0779	0780	0781	0782	0783
31_	0784	0785	0786	0787	0788	0789	0790	0791	0792	0793	0794	0795	0796	0797	0798	0799
32_	0800	0801	0802	0803	0804	0805	0806	0807	0808	0809	0810	0811	0812	0813	0814	0815
33_	0816	0817	0818	0819	0820	0821	0822	0823	0824	0825	0826	0827	0828	0829	0830	0831
34_	0832	0833	0834	0835	0836	0837	0838	0839	0840	0841	0842	0843	0844	0845	0846	0847
35_	0848	0849	0850	0851	0852	0853	0854	0855	0856	0857	0858	0859	0860	0861	0862	0863
36_	0864	0865	0866	0867	0868	0869	0870	0871	0872	0873	0874	0875	0876	0877	0878	0879
37_	0880	0881	0882	0883	0884	0885	0886	0887	0888	0889	0890	0891	0892	0893	0894	0895
38_	0896	0897	0898	0899	0900	0901	0902	0903	0904	0905	0906	0907	0908	0909	0910	0911
39_	0912	0913	0914	0915	0916	0917	0918	0919	0920	0921	0922	0923	0924	0925	0926	0927
3A_	0928	0929	0930	0931	0932	0933	0934	0935	0936	0937	0938	0939	0940	0941	0942	0943
3B_	0944	0945	0946	0947	0948	0949	0950	0951	0952	0953	0954	0955	0956	0957	0958	0959
3C_	0960	0961	0962	0963	0964	0965	0966	0967	0968	0969	0970	0971	0972	0973	0974	0975
3D_	0976	0977	0978	0979	0980	0981	0982	0983	0984	0985	0986	0987	0988	0989	0990	0991
3E_	0992	0993	0994	0995	0996	0997	0998	0999	1000	1001	1002	1003	1004	1005	1006	1007
3F_	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
40_	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039
41_	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055
42_	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071
43_	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087
44_	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103
45_	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119
46_	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135
47_	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151
48_	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167
49_	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183
4A_	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199
4B_	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215
4C_	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231
4D_	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247
4E_	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263
4F_	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279
50_	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295
51_	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311
52_	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327
53_	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343
54_	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359
55_	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375
56_	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391
57_	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407
58_	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423
59_	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439
5A_	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455
5B_	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471
5C_	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487
5D_	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503
5E_	1504	1505	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1516	1517	1518	1519
5F_	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530	1531	1532	1533	1534	1535

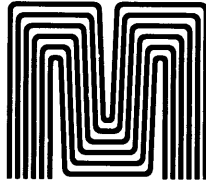
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
60_	1536	1537	1538	1539	1540	1541	1542	1543	1544	1545	1546	1547	1548	1549	1550	1551
61_	1552	1553	1554	1555	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567
62_	1568	1569	1570	1571	1572	1573	1574	1575	1576	1577	1578	1579	1580	1581	1582	1583
63_	1584	1585	1586	1587	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599
64_	1600	1601	1602	1603	1604	1605	1606	1607	1608	1609	1610	1611	1612	1613	1614	1615
65_	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631
66_	1632	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647
67_	1648	1649	1650	1651	1652	1653	1654	1655	1656	1657	1658	1659	1660	1661	1662	1663
68_	1664	1665	1666	1667	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679
69_	1680	1681	1682	1683	1684	1685	1686	1687	1688	1689	1690	1691	1692	1693	1694	1695
6A_	1696	1697	1698	1699	1700	1701	1702	1703	1704	1705	1706	1707	1708	1709	1710	1711
6B_	1712	1713	1714	1715	1716	1717	1718	1719	1720	1721	1722	1723	1724	1725	1726	1727
6C_	1728	1729	1730	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743
6D_	1744	1745	1746	1747	1748	1749	1750	1751	1752	1753	1754	1755	1756	1757	1758	1759
6E_	1760	1761	1762	1763	1764	1765	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775
6F_	1776	1777	1778	1779	1780	1781	1782	1783	1784	1785	1786	1787	1788	1789	1790	1791
70_	1792	1793	1794	1795	1796	1797	1798	1799	1800	1801	1802	1803	1804	1805	1806	1807
71_	1808	1809	1810	1811	1812	1813	1814	1815	1816	1817	1818	1819	1820	1821	1822	1823
72_	1824	1825	1826	1827	1828	1829	1830	1831	1832	1833	1834	1835	1836	1837	1838	1839
73_	1840	1841	1842	1843	1844	1845	1846	1847	1848	1849	1850	1851	1852	1853	1854	1855
74_	1856	1857	1858	1859	1860	1861	1862	1863	1864	1865	1866	1867	1868	1869	1870	1871
75_	1872	1873	1874	1875	1876	1877	1878	1879	1880	1881	1882	1883	1884	1885	1886	1887
76_	1888	1889	1890	1891	1892	1893	1894	1895	1896	1897	1898	1899	1900	1901	1902	1903
77_	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913	1914	1915	1916	1917	1918	1919
78_	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935
79_	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1949	1950	1951
7A_	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967
7B_	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983
7C_	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
7D_	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
7E_	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031
7F_	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
80_	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063
81_	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079
82_	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095
83_	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111
84_	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127
85_	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143
86_	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159
87_	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175
88_	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191
89_	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207
8A_	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223
8B_	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239
8C_	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255
8D_	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271
8E_	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287
8F_	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303
90_	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319
91_	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335
92_	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351
93_	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367
94_	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383
95_	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399
96_	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415
97_	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431
98_	2432	2433	2434	2435	2436	2437	2438	2439	2440	2441	2442	2443	2444	2445	2446	2447
99_	2448	2449	2450	2451	2452	2453	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463
9A_	2464	2465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475	2476	2477	2478	2479
9B_	2480	2481	2482	2483	2484	2485	2486	2487	2488	2489	2490	2491	2492	2493	2494	2495
9C_	2496	2497	2498	2499	2500	2501	2502	2503	2504	2505	2506	2507	2508	2509	2510	2511
9D_	2512	2513	2514	2515	2516	2517	2518	2519	2520	2521	2522	2523	2524	2525	2526	2527
9E_	2528	2529	2530	2531	2532	2533	2534	2535	2536	2537	2538	2539	2540	2541	2542	2543
9F_	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553	2554	2555	2556	2557	2558	2559

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A0_	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569	2570	2571	2572	2573	2574	2575
A1_	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585	2586	2587	2588	2589	2590	2591
A2_	2592	2593	2594	2595	2596	2597	2598	2599	2600	2601	2602	2603	2604	2605	2606	2607
A3_	2608	2609	2610	2611	2612	2613	2614	2615	2616	2617	2618	2619	2620	2621	2622	2623
A4_	2624	2625	2626	2627	2628	2629	2630	2631	2632	2633	2634	2635	2636	2637	2638	2639
A5_	2640	2641	2642	2643	2644	2645	2646	2647	2648	2649	2650	2651	2652	2653	2654	2655
A6_	2656	2657	2658	2659	2660	2661	2662	2663	2664	2665	2666	2667	2668	2669	2670	2671
A7_	2672	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686	2687
A8_	2688	2689	2690	2691	2692	2693	2694	2695	2696	2697	2698	2699	2700	2701	2702	2703
A9_	2704	2705	2706	2707	2708	2709	2710	2711	2712	2713	2714	2715	2716	2717	2718	2719
AA_	2720	2721	2722	2723	2724	2725	2726	2727	2728	2729	2730	2731	2732	2733	2734	2735
AB_	2736	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	2748	2749	2750	2751
AC_	2752	2753	2754	2755	2756	2757	2758	2759	2760	2761	2762	2763	2764	2765	2766	2767
AD_	2768	2769	2770	2771	2772	2773	2774	2775	2776	2777	2778	2779	2780	2781	2782	2783
AE_	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794	2795	2796	2797	2798	2799
AF_	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	2813	2814	2815
B0_	2816	2817	2818	2819	2820	2821	2822	2823	2824	2825	2826	2827	2828	2829	2830	2831
B1_	2832	2833	2834	2835	2836	2837	2838	2839	2840	2841	2842	2843	2844	2845	2846	2847
B2_	2848	2849	2850	2851	2852	2853	2854	2855	2856	2857	2858	2859	2860	2861	2862	2863
B3_	2864	2865	2866	2867	2868	2869	2870	2871	2872	2873	2874	2875	2876	2877	2878	2879
B4_	2880	2881	2882	2883	2884	2885	2886	2887	2888	2889	2890	2891	2892	2893	2894	2895
B5_	2896	2897	2898	2899	2900	2901	2902	2903	2904	2905	2906	2907	2908	2909	2910	2911
B6_	2912	2913	2914	2915	2916	2917	2918	2919	2920	2921	2922	2923	2924	2925	2926	2927
B7_	2928	2929	2930	2931	2932	2933	2934	2935	2936	2937	2938	2939	2940	2941	2942	2943
B8_	2944	2945	2946	2947	2948	2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959
B9_	2960	2961	2962	2963	2964	2965	2966	2967	2968	2969	2970	2971	2972	2973	2974	2975
BA_	2976	2977	2978	2979	2980	2981	2982	2983	2984	2985	2986	2987	2988	2989	2990	2991
BB_	2992	2993	2994	2995	2996	2997	2998	2999	3000	3001	3002	3003	3004	3005	3006	3007
BC_	3008	3009	3010	3011	3012	3013	3014	3015	3016	3017	3018	3019	3020	3021	3022	3023
BD_	3024	3025	3026	3027	3028	3029	3030	3031	3032	3033	3034	3035	3036	3037	3038	3039
BE_	3040	3041	3042	3043	3044	3045	3046	3047	3048	3049	3050	3051	3052	3053	3054	3055
BF_	3056	3057	3058	3059	3060	3061	3062	3063	3064	3065	3066	3067	3068	3069	3070	3071

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C0_	3072	3073	3074	3075	3076	3077	3078	3079	3080	3081	3082	3083	3084	3085	3086	3087
C1_	3088	3089	3090	3091	3092	3093	3094	3095	3096	3097	3098	3099	3100	3101	3102	3103
C2_	3104	3105	3106	3107	3108	3109	3110	3111	3112	3113	3114	3115	3116	3117	3118	3119
C3_	3120	3121	3122	3123	3124	3125	3126	3127	3128	3129	3130	3131	3132	3133	3134	3135
C4_	3136	3137	3138	3139	3140	3141	3142	3143	3144	3145	3146	3147	3148	3149	3150	3151
C5_	3152	3153	3154	3155	3156	3157	3158	3159	3160	3161	3162	3163	3164	3165	3166	3167
C6_	3168	3169	3170	3171	3172	3173	3174	3175	3176	3177	3178	3179	3180	3181	3182	3183
C7_	3184	3185	3186	3187	3188	3189	3190	3191	3192	3193	3194	3195	3196	3197	3198	3199
C8_	3200	3201	3202	3203	3204	3205	3206	3207	3208	3209	3210	3211	3212	3213	3214	3215
C9_	3216	3217	3218	3219	3220	3221	3222	3223	3224	3225	3226	3227	3228	3229	3230	3231
CA_	3232	3233	3234	3235	3236	3237	3238	3239	3240	3241	3242	3243	3244	3245	3246	3247
CB_	3248	3249	3250	3251	3252	3253	3254	3255	3256	3257	3258	3259	3260	3261	3262	3263
CC_	3264	3265	3266	3267	3268	3269	3270	3271	3272	3273	3274	3275	3276	3277	3278	3279
CD_	3280	3281	3282	3283	3284	3285	3286	3287	3288	3289	3290	3291	3292	3293	3294	3295
CE_	3296	3297	3298	3299	3300	3301	3302	3303	3304	3305	3306	3307	3308	3309	3310	3311
CF_	3312	3313	3314	3315	3316	3317	3318	3319	3320	3321	3322	3323	3324	3325	3326	3327
D0_	3328	3329	3330	3331	3332	3333	3334	3335	3336	3337	3338	3339	3340	3341	3342	3343
D1_	3344	3345	3346	3347	3348	3349	3350	3351	3352	3353	3354	3355	3356	3357	3358	3359
D2_	3360	3361	3362	3363	3364	3365	3366	3367	3368	3369	3370	3371	3372	3373	3374	3375
D3_	3376	3377	3378	3379	3380	3381	3382	3383	3384	3385	3386	3387	3388	3389	3390	3391
D4_	3392	3393	3394	3395	3396	3397	3398	3399	3400	3401	3402	3403	3404	3405	3406	3407
D5_	3408	3409	3410	3411	3412	3413	3414	3415	3416	3417	3418	3419	3420	3421	3422	3423
D6_	3424	3425	3426	3427	3428	3429	3430	3431	3432	3433	3434	3435	3436	3437	3438	3439
D7_	3440	3441	3442	3443	3444	3445	3446	3447	3448	3449	3450	3451	3452	3453	3454	3455
D8_	3456	3457	3458	3459	3460	3461	3462	3463	3464	3465	3466	3467	3468	3469	3470	3471
D9_	3472	3473	3474	3475	3476	3477	3478	3479	3480	3481	3482	3483	3484	3485	3486	3487
DA_	3488	3489	3490	3491	3492	3493	3494	3495	3496	3497	3498	3499	3500	3501	3502	3503
DB_	3504	3505	3506	3507	3508	3509	3510	3511	3512	3513	3514	3515	3516	3517	3518	3519
DC_	3520	3521	3522	3523	3524	3525	3526	3527	3528	3529	3530	3531	3532	3533	3534	3535
DD_	3536	3537	3538	3539	3540	3541	3542	3543	3544	3545	3546	3547	3548	3549	3550	3551
DE_	3552	3553	3554	3555	3556	3557	3558	3559	3560	3561	3562	3563	3564	3565	3566	3567
DF_	3568	3569	3570	3571	3572	3573	3574	3575	3576	3577	3578	3579	3580	3581	3582	3583

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E0_	3584	3585	3586	3587	3583	3589	3590	3591	3592	3593	3594	3595	3596	3597	3598	3599
E1_	3600	3601	3602	3603	3604	3605	3606	3607	3608	3609	3610	3611	3612	3613	3614	3615
E2_	3616	3617	3618	3619	3620	3621	3622	3623	3624	3625	3626	3627	3628	3629	3630	3631
E3_	3632	3633	3634	3635	3636	3637	3638	3639	3640	3641	3642	3643	3644	3645	3646	3647
E4_	3648	3649	3650	3651	3652	3653	3654	3655	3656	3657	3658	3659	3660	3661	3662	3663
E5_	3664	3665	3666	3667	3668	3669	3670	3671	3672	3673	3674	3675	3676	3677	3678	3679
E6_	3680	3681	3682	3683	3684	3685	3686	3687	3688	3689	3690	3691	3692	3693	3694	3695
E7_	3696	3697	3698	3699	3700	3701	3702	3703	3704	3705	3706	3707	3708	3709	3710	3711
E8_	3712	3713	3714	3715	3716	3717	3718	3719	3720	3721	3722	3723	3724	3725	3726	3727
E9_	3728	3729	3730	3731	3732	3733	3734	3735	3736	3737	3738	3739	3740	3741	3742	3743
EA_	3744	3745	3746	3747	3748	3749	3750	3751	3752	3753	3754	3755	3756	3757	3758	3759
EB_	3760	3761	3762	3763	3764	3765	3766	3767	3768	3769	3770	3771	3772	3773	3774	3775
EC_	3776	3777	3778	3779	3780	3781	3782	3783	3784	3785	3786	3787	3788	3789	3790	3791
ED_	3792	3793	3794	3795	3796	3797	3798	3799	3800	3801	3802	3803	3804	3805	3806	3807
EE_	3808	3809	3810	3811	3812	3813	3814	3815	3816	3817	3818	3819	3820	3821	3822	3823
EF_	3824	3825	3826	3827	3828	3829	3830	3831	3832	3833	3834	3835	3836	3837	3838	3839
F0_	3840	3841	3842	3843	3844	3845	3846	3847	3848	3849	3850	3851	3852	3853	3854	3855
F1_	3856	3857	3858	3859	3860	3861	3862	3863	3864	3865	3866	3867	3868	3869	3870	3871
F2_	3872	3873	3874	3875	3876	3877	3878	3879	3880	3881	3882	3883	3884	3885	3886	3887
F3_	3888	3889	3890	3891	3892	3893	3894	3895	3896	3897	3898	3899	3900	3901	3902	3903
F4_	3904	3905	3906	3907	3908	3909	3910	3911	3912	3913	3914	3915	3916	3917	3918	3919
F5_	3920	3921	3922	3923	3924	3925	3926	3927	3928	3929	3930	3931	3932	3933	3934	3935
F6_	3936	3937	3938	3939	3940	3941	3942	3943	3944	3945	3946	3947	3948	3949	3950	3951
F7_	3952	3953	3954	3955	3956	3957	3958	3959	3960	3961	3962	3963	3964	3965	3966	3967
F8_	3968	3969	3970	3971	3972	3973	3974	3975	3976	3977	3978	3979	3980	3981	3982	3983
F9_	3984	3985	3986	3987	3988	3989	3990	3991	3992	3993	3994	3995	3996	3997	3998	3999
FA_	4000	4001	4002	4003	4004	4005	4006	4007	4008	4009	4010	4011	4012	4013	4014	4015
FB_	4016	4017	4018	4019	4020	4021	4022	4023	4024	4025	4026	4027	4028	4029	4030	4031
FC_	4032	4033	4034	4035	4036	4037	4038	4039	4040	4041	4042	4043	4044	4045	4046	4047
FD_	4048	4049	4050	4051	4052	4053	4054	4055	4056	4057	4058	4059	4060	4061	4062	4063
FE_	4064	4065	4066	4067	4068	4069	4070	4071	4072	4073	4074	4075	4076	4077	4078	4079
FF_	4080	4081	4082	4083	4084	4085	4086	4087	4088	4089	4090	4091	4092	4093	4094	4095



Micro Systems Inc.

COMMENT AND EVALUATION SHEET MICRO 810 COMPUTER AP810 Assembly Program

Pub. No. 69-2-0810-002

January 1970

YOUR EVALUATION OF THIS MANUAL WILL BE WELCOMED BY
MICRO SYSTEMS INC. ANY ERRORS, SUGGESTED ADDITIONS OR
DELETIONS OR GENERAL COMMENTS MAY BE MADE BELOW.
PLEASE INCLUDE PAGE NUMBER REFERENCE.

FROM NAME: _____

BUSINESS
ADDRESS: _____

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

FOLD ON DOTTED LINES AND STAPLE

STAPLE

STAPLE

FOLD

FOLD

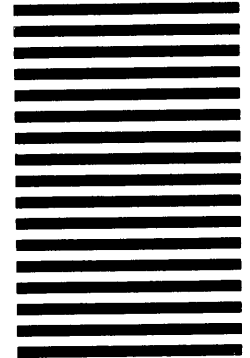
First Class
Permit No. 1972
Santa Ana
California 92711

BUSINESS REPLY MAIL
NO POSTAGE NECESSARY IF MAILED IN THE U.S.A.

Postage Will Be Paid By

MICRO SYSTEMS INC.

644 East Young Street
Santa Ana, California 92705

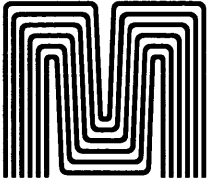


FOLD

FOLD

STAPLE

STAPLE



MicroSystems Inc.

A Microdata Subsidiary

644 East Young Street
Santa Ana, California 92705
Telephone: (714) 540-6730
TWX: 910-595-1764