SECTION 2
INTRODUCTION TO AIM 65 OPERATION

This section presents the basic methods of AIM 65 operation. We will describe, in order, how to:

1. Examine the contents of a memory location.
2. Change the contents of a memory location.
3. Enter a machine language program into memory.
4. Enter data into memory.
5. Execute a machine language program.
6. Examine the contents of a register.
7. Change the contents of a register.
8. Use the printer.
9. Store a program or data on a cassette.
10. Load a program or data from a cassette.

Section 3 describes all of these operations in greater detail. Section 3 also discusses other features of the AIM 65 that are used in debugging, in conjunction with the optional assembler and BASIC, and in actual applications.

Let us first discuss how to get started. We assume that you have connected and checked the AIM 65 as discussed in Section 1. Press the RESET button which is located just below and to the left of the printer. You should now be ready to start operating the AIM 65. If you become confused or lost at any time, press RESET. This will return control to the monitor. If you just wish to terminate an operation, type ESC; this key concludes most simple monitor commands.

For the first part of this discussion, start the AIM 65 in the following state:

1. Printer off. You can toggle the printer (i.e. turn it on if it is off and off if it is on) by pressing CTRL and PRINT together. Note that the AIM 65 displays the current state of the printer. Type RETURN after you turn the printer off. We will explain how to use the printer later in this section.

2. KB/TTY switch in the KB (user keyboard) position.

3. RUN/STEP switch in the RUN position.

These switches are located just to the left of the display.

Table 2-1. Hexadecimal to Decimal Conversion

HEXADECIMAL DIGIT	DECIMAL VALUE	BINARY VALUE
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

2.1 EXAMINING MEMORY

You may examine the contents of memory by typing M and entering the hexadecimal address that you want to examine. Table 2-1 contains a list of the hexadecimal digits and their decimal and binary equivalents.

For example, type:

M

0

RETURN

to display the contents of memory addresses 0, 1, 2, and 3 from left to right. Note that the AIM 65 prompts you to enter an address after you type M.

Now type a space. The AIM 65 responds by displaying the contents of the next four memory addresses. Note that the AIM always displays the starting address to the left of the data. You can continue typing spaces and examining memory, four locations at a time.

Note the following features of the memory examination:

1. Memory addresses are 4 hexadecimal digits long (16 bits) while the contents of a memory location are 2 hexadecimal digits long (8 bits).

2. You can move forward in memory (by pressing the space bar) but not back.

3. All addresses and data are displayed in hexadecimal. If you are unfamiliar with this number system, refer to Table 2-1. There are also explanations of hexadecimal numbers in many books, including T. C. Bartee, Digital Computer Fundamentals, McGraw-Hill, New York, 1974.

Note that you can start examining memory at any address. For example, type:

M

F

8

A

6

RETURN

to display the contents of memory locations F8A6, F8A7, F8A8, and F8A9 from left to right.

2.2 CHANGING MEMORY

You may change the contents of a memory location (after you have examined it) by typing / and then entering the new contents in hexadecimal (2 digits).

For example, examine the contents of memory locations 0, 1, 2, and 3 by typing:

M

0

RETURN

Now you can change the contents of memory location 0 to 05 by typing:

/

0

5

RETURN

Note that the AIM 65 prompts you by displaying the starting address after you type /. You can check to be sure that the memory was actually changed by repeating the examination procedure, i.e. by typing:

M

0

RETURN

Memory location 0 should now contain 05. Be careful--you must enter 05, not just 5. What happens if you type /, 5, RETURN?

In fact, you can change any or all of the four displayed locations in one operation by typing the new values in order and typing spaces for any values that you want to leave unchanged.

For example, let us place 06 in memory location 0, B7 in memory location 1, and E3 in memory location 3 while leaving memory location 2 unchanged. Type:

M	EXAMINE LOCATION
0	
RETURN	
/	(0) = 06
0	
6	
B	(1) = B7
7	
SPACE	(2) unchanged
E	(3) = E3
3	

We use parentheses around a memory address to indicate "contents of", i.e., (5) refers to the 8-bit data located at memory address 5.

Note that the AIM 65 prompts you by moving its cursor across the display. If you change or space over the last memory location no RETURN is necessary.

If you want to continue changing the contents of memory just press / again. The AIM 65 will display the next higher address. Remember that you can use M and SPACE to check to see that you have entered the data properly.

2.3 ENTERING A PROGRAM

To enter a machine language program, you must use the R6500 Microprocessor Programming Card. This card contains the 3-letter mnemonics (see Table 5-2) for all R6500 instructions. These instructions and R6500 assembly language programming are described in more detail in Section 6 of this manual and in the R6500 Programming Manual. For now, we will just discuss some simple examples.

You can enter a program by typing I followed by the proper series of mnemonics and operands. The operands give the microprocessor the additional information that it needs to execute the instructions (e.g., the memory address from which to load the accumulator or the destination for a branch instruction). Some instructions like TAX (move A to X) or CLC (clear carry) need no operands since the processor knows what to do from the operation code alone. On the Reference Card, such instructions are described as having implied addressing.

Let us look at a simple example program that logically ANDs the contents of memory locations 40 and 41 and places the result in memory location 42. Remember that all the addresses are hexadecimal. The program is:

```
LDA     40

AND     41

STA     42

BRK
```

Note the following features of this program:

1. LDA 40 loads the accumulator from memory location 40. The address is really 0040 but we do not have to enter the leading zeros.

2. AND 41 logically ANDs the accumulator with the contents of memory location 41. The result is placed in the accumulator.

3. STA 42 stores the accumulator in memory location 42.

4. BRK returns control to the AIM 65 monitor after the program has been executed. You should place this instruction at the end of all your programs so that the computer does not go wandering off aimlessly. Remember that the computer will continue executing instructions sequentially unless it is specifically told to do otherwise.

Now let us enter the program into memory as follows:

1. Type I. The AIM 65 responds by displaying the memory address at which it will start placing the instructions.

2. If the starting address is not zero, type *, 0, RETURN to make it zero.

3. Type L, D, A, 4, 0, SPACE to enter the LDA 40 instruction. Note that the AIM 65 automatically displays the memory address in which the next instruction will be placed.

4. Type A, N, D, 4, 1, SPACE to enter the AND 41 instruction.

5. Type S, T, A, 4, 2, SPACE to enter the STA 42 instruction.

6. Type B, R, K to enter the BRK instruction. Note that no SPACE is necessary since the BRK instruction requires no operands.

7. Type ESC to end program entry.

If you make a mistake, you can generally recover quite easily. In fact, the AIM 65 simply ignores most typing errors such as SAT instead of STA. The problems come when you accidently type a valid code that is not the one you wanted (like STX instead of STA) or type an address incorrectly (e.g., 43 instead of 42).

If you catch the error before you complete the mnemonic code or type RETURN, you can backspace and erase by typing DEL. Note that a character disappears from the display and the marker moves backward. However, this does not work if you have entered a 3-letter mnemonic or typed RETURN. Then you must correct the line by restarting the entry procedure at the address where you made the error.

For example, if I typed STX 42 instead of STA 42 at address 4, I could correct my error by typing:

```
*           AT ADDRESS 4

4

RETURN

S           STA 42

T

A

4

2

SPACE
```

Note that all we have done so far is enter the program into memory. We have not yet entered any data, executed the program, or produced any results.

Still another simple program takes the contents of memory location 40, clears the four most significant bits, and stores the result in memory location 41. We can clear the four most significant bits by logically ANDing the accumulator with 0F hex (00001111). Remember that logically ANDing with a '0' always gives zero (why?). The program is:

```
LDA     40

AND     #0F

STA     41

BRK
```

Note the following features of this program:

1. AND #0F logically ANDs the accumulator with the number 0F. This is called immediate addressing. Note the difference between AND #0F and AND 0F which logically ANDs the accumulator with the contents of memory location 000F. That memory location could contain any 8-bit number.

2. The '#' sign means "immediate', i.e. the following number is data rather than an address.

3. The BRK instruction at the end of the program restores control to the monitor just as in the previous example.

We can enter this program as follows:

1. Type I

2. If necessary, type *, 0, RETURN to make the starting address zero.

3.. Type L, D, A, 4, 0, SPACE to enter the LDA 40 instruction.

4. Type A, N, D, #, 0, F, SPACE to enter the AND # OF instruction. Remember to shift to type "#".

5. Type S, T, A, 4, 1, SPACE to enter the STA 41 instruction.

6. Type B, R, K to enter the BRK instruction.

7. Type ESC to end program entry.

You should read the description of the I command in Section 3.5.1 for details on how to enter instructions that we have not discussed here.

2.4 ENTERING DATA

Before we have the AIM 65 execute a program, we need some way of entering data and observing the results. This is simple since we can use the procedures that we have described previously for examining and changing the contents of memory.

For example, the first program from the previous discussion was:

```
LDA     40
AND     41
STA     42
BRK
```

This program requires data in memory locations 40 and 41. The result is saved in memory location 42.

Entering the data requires the following steps:

1. Type M, 4, 0, RETURN to observe the contents of memory locations 40 through 43.

2. Type /, B, 7, 6, 3, RETURN to enter the data into memory locations 40 and 41. We have placed B7 in memory location 40 and 63 in memory location 41 but any other values would be just as easy to enter.

Note that you might want to put zero in memory location 42 just to be sure that the answer was not already there.

To observe the result after the program has been executed, all we have to do is type M, 4, 0, RETURN. The first two numbers are the original data (memory locations 40 and 41) while the third number is the result (memory location 42). The situation is even simpler for the second example program since it only uses memory locations 40 (for the original data) and 41 (for the result).

2.5 EXECUTING A PROGRAM

To have the AIM 65 execute a program all we have to do is tell it where to start and then use the G command (for GO). Remember to put a BRK instruction at the end of your program or the AIM 65 may go and never come back. If this happens, press the RESET button.

So, to have the AIM 65 execute a program starting in memory location 0, set the RUN/STEP switch to RUN and type:

Key	Comment
*	STARTING ADDRESS
0	
RETURN	
G	GO
RETURN	

Note that typing G is just one step in a long process. To actually run a program we must:

1. Enter the program into memory using the I command.

2. Enter the data into memory using the M and / commands.

3. Execute the program using the G command.

4. Observe the results using the M command.

Let us now see how the entire procedure works in some simple cases.

Example:

Logically AND the contents of memory locations 40 and place the result in 42.

DATA:

(40)=B7
(41)=63

RESULT:

(42)=23

Remember that the parentheses around the address means "contents of".

1. PROGRAM ENTRY

Type

I	
*	BEGIN PROGRAM ENTRY
0	AT ADDRESS 0
RETURN	
L	LDA 40
D	
A	
4	
0	
SPACE	

A	AND 41
N	
D	
4	
1	
SPACE	
S	STA 42
T	
A	
4	
2	
SPACE	BRK
B	
R	
K	
ESC	END PROGRAM

2. DATA ENTRY

Type

M	EXAMINE MEMORY
4	AT ADDRESS 0
0	
RETURN	
/	CHANGE MEMORY
B	(40)=B7
7	
6	(41)=63
3	
0	(42)=00
0	
RETURN	

3. PROGRAM EXECUTION

Type

*	STARTING ADDRESS=0
0	
RETURN	
G	GO
RETURN	

4. OBSERVING RESULTS

Type

M	EXAMINE MEMORY
4	AT ADDRESS 40
0	
RETURN	

The result is the third number.

Try going through this procedure once. Repeat it for the following sample cases.

A. (40)=F3
(41)=9A

Result = (42)=92

B. (40)=D7
(41)=AB

Result = (42)=83

Example:

Clear the four most significant bits of memory location 40 and place the result in memory location 41.

DATA:

(40)=B7

RESULT:

(41)=07

1. PROGRAM ENTRY

Type

I	BEGIN PROGRAM ENTRY
*	AT ADDRESS 0
0	
RETURN	
L	LDA 40
D	
A	
4	
0	
SPACE	
A	AND #0F
N	
D	
#	
0	
F	

SPACE	
S	
T	
A	
4	
1	
SPACE	
B	BRK
R	
K	
ESC	END PROGRAM ENTRY

2. DATA ENTRY

Type	
M	EXAMINE MEMORY
4	AT ADDRESS 40
0	
RETURN	
/	
B	(40)=B7
7	
0	(41)=00
0	
RETURN	

3. PROGRAM EXECUTION

Type	
*	STARTING ADDRESS = 0
0	
RETURN	
G	GO
RETURN	

4. OBSERVING RESULTS

Type	
M	EXAMINE MEMORY
4	AT ADDRESS 41
1	
RETURN	

Result = (41) = 07

Try going through this procedure once. Repeat it for the following sample cases:

A. (40)=F3

Result = (41)=03

B. (40)=AB

Result = (41)=0B

2.6 EXAMINING REGISTERS

The R6502 microprocessor actually performs its operations using the following registers:

Program Counter
Processor Status or P register
Accumulator or A register
Index register X or X register
Index register Y or Y register
Stack pointer or S register

Let us now briefly discuss each of these registers. There is a more complete description in the R6500 Programming Manual.

1. PROGRAM COUNTER (or PC)

This is a 16-bit register which holds the address of the next instruction to be executed. Every time the processor uses this register, it adds one to the contents. Thus, the processor executes instructions sequentially unless a JUMP or BRANCH instruction specifically places a new value in the program counter.

2. PROCESSOR STATUS (or P)

This is an 8 bit register which reflects the current status of the CPU. Its bits are: (See Figure 2-1.)

Bit 7 (N)=1 if the last result had a 1 in its most significant bit, 0 if the last result had a 0 in its most significant bit. This bit is often called the NEGATIVE or SIGN flag.

Bit 6 (V)=1 if the last arithmetic operation produced a two's complement overflow, 0 if it did not. This bit is called the OVERFLOW flag.

Bit 5 = not used.

Bit 4 (B)=1 if the last instruction was BRK, 0 otherwise. This bit is called the BREAK COMMAND flag.

Bit 3 (D)=1 if the processor is in decimal mode, 0 if it is not. The bit is called the DECIMAL MODE flag.

Bit 2 (I)=1 if interrupts are <u>not</u> allowed, 0 if they are. This bit is called INTERRUPT DISABLE flag..

Bit 1 (Z)=1 if the last result <u>was zero</u>, 0 if it was not. This bit is called the ZERO flag.

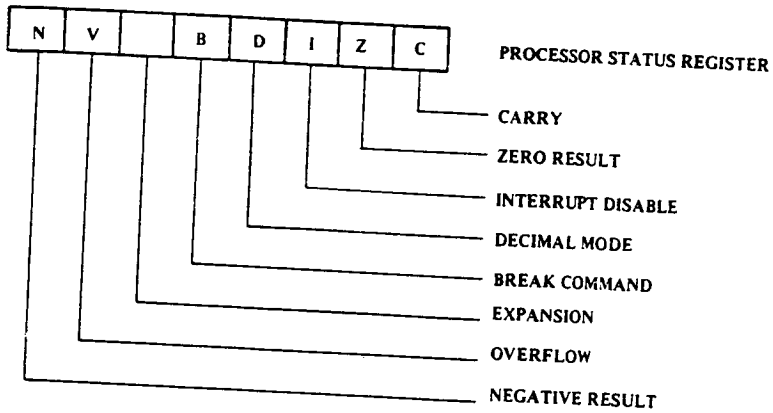

Figure 2-1. Processor Status Register

Bit 0 (C)=1 if the last addition produced a carry or the last subtraction did not require a borrow, 0 if the opposite conditions held. This bit is called the CARRY flag.

NOTE

Only the individual bits in the P register are meaningful. If you wish to observe or change those bits, you should consult Table 2-1 to convert between binary and hexadecimal.

3. ACCUMULATOR (or A)

This is an 8 bit register which is the center of processor operations. It acts much like the current sub-total in a calculator.

4/5. INDEX REGISTERS X and Y

These are two 8-bit registers which can be used as counters or indexes.

6. STACK POINTER (or S)

This is an 8-bit register which contains the address of the stack on page 1 of memory. If S contains F3, the next available stack location is at address 01F3.

To observe the current contents of all registers, type R. The AIM 65 will display the registers in the following order.

PC P A X Y S

Note that the program counter is 4 digits long while the other registers are 2 digits long.

2.7 CHANGING REGISTERS

You may change the contents of the registers with the following commands. Remember that PC is 4 digits long:

1. PC - *
2. Accumulator -A
3. X register -X
4. Y register -Y
5. Stack pointer -S
6. Processor status -P

We have listed these roughly in the order of frequency of use. You will find that you often want to change the program counter, accumulator and index registers. You will seldom want to change the stack pointer or processor status.

Examples:

1. Place 03E1 in the Program Counter.

```
Type
*               ALTER PC

0               (PC)=03E1
3
E
1
RETURN
```

2. Place 5F in the accumulator.

```
Type

A               ALTER A
5               (A)=5F
F
```


3. Place 10 in index register X.

```
Type

X               ALTER X
1               (X)=10
0
```

4. Place 3 in index register Y.

```
Type

Y               ALTER Y
3               (Y)=37
7
```

Remember that all entries are in hexadecimal.

2.8 USING THE PRINTER

You can control the printer as follows:

1. Press CTRL and PRINT simultaneously to turn the printer on if it is off, and off if it is on. Note that the displays tell you the current state of the printer.

2. Press PRINT to have the printer print whatever is on the display. This command works even if you have turned the printer off.

3. Press LF (line feed) to advance the paper.

The printer output can give you a permanent record to study or retain. But there is no use wasting a lot of paper if you are just trying things out or checking operations.

2.9 RECORDING ON CASSETTES

Once you have entered a program into memory and corrected it you will probably want to record it on a cassette rather than re-enter it each time from the keyboard. We assume that the audio cassette recorder has been previously attached in position 1 according to the instructions in Section 9. We also assume that the volume control has been set appropriately (usually to the highest level).

The following procedure will allow you to record your program on tape.

1. Place a cassette in the recorder, rewind it, and then play it until you are past the leader. If your recorder has a counter, allow at least five counts.

2. Type D (for dump).

3. In response to the AIM 65 displaying FROM =, type the starting address of the program followed by RETURN.

4. In response to the AIM 65 displaying TO =, type the ending address of the program followed by RETURN.

5. In response to the AIM 65 displaying OUT =, type the device code T for audio tape in AIM 65 format.

6. In response to the AIM 65 displaying F =, type the file name. This can be any five alphanumeric or special characters. If the name is less than five characters long, type SPACE at the end. Simple file names would be PROG1, A1P6, SUM, or TEST.

7. In response to the AIM 65 displaying T =, type the recorder number (1).

8. Place the recorder in the record mode by pressing PLAY and RECORD simultaneously.

9. Type RETURN. The AIM 65 will now record the program on tape.

10. In response to the AIM 65 displaying MORE?, type N (for NO). The AIM 65 will then complete the recording.

11. When the recording has been completed, stop the recorder.

2.10 LOADING FROM CASSETTES

To load a program from a cassette, use the following procedure. We again assume that the audio cassette recorder has been attached according to the instructions in Section 9 and that the volume control has been set appropriately.

1. Place the cassette in the recorder and rewind it.

2. Type L (for load).

3. In response to the AIM 65 displaying IN =, type the device code T for audio tape in AIM 65 format.

4. In response to the AIM 65 displaying F =, enter the file name that you used in recording the tape. Type SPACE at the end if the name is less than five characters long.

5. In response to the AIM 65 displaying T =, type the recorder number (1).

6. Place the recorder in the Read Mode by pressing PLAY.

7. Type RETURN. The AIM 65 will now load the program from tape into memory.

8. When the program has been completely loaded, turn the recorder off.