Introducing Microprocessors Part 4

More on system monitors, plus an assessment quiz

MIKE TOOLEY

In Part Three we mentioned the use of a monitor program as an aid to understanding the operation of a microprocessor. We also stated that such a program can also allow us to enter, test and debug simple programs. In small microcomputer systems (of the sort used for education purposes) monitor programs are invariably provided in ROM. In other systems (such as the IBM PC and compatibles) the monitor program is resident in RAM and must be loaded from disk. Naturally, with more powerful and more modern systems, monitor programs offer a number of extended facilities. These programs (usually referred to as "debuggers") are often provided as an extension of the operating system, and their name might imply, they can be an invaluable aid for programmers and software developers. One of the most popular debuggers is that which runs under the MS-DOS operating system on the IBM PC (and compatibles). This program is DEBUG.COM.

For the purpose of this series, we shall assume that readers only have access to a fairly basic monitor and we shall discuss each of the features of a monitor with which readers should be familiar. Such a discussion will obviously be more meaningful if readers have reasonably immediate access to a microprocessor system and can try out the various monitor commands (usually single letters followed by one or more parameters) as they are introduced.

Memory Display

A memory display facility can be used to display the contents of a given memory address or range of addresses. The display is usually presented in hexadecimal format though systems which provide output to a TV, monitor, or VDU terminal generally also provide an ASCII representation of the data. This latter facility can be useful when the investigation contains text rather than a machine code program.
A typical memory display command would take the form:

D 4000 (Display an 80 byte block of memory starting at address 4000 hex.)

A typical memory display (in hexadecimal and ASCII) is shown in Fig. 4.1. Note that some monitors use a Memory Pointer which must be set before the contents of a given block of memory can be displayed. A typical sequence of commands would then be:

N 4000 (Set memory pointer to 4000 hex.)
L (List 80 bytes of memory starting from 4000 hex.)

Other monitors require that start and end addresses are specified as part of the command as shown in the following command which prints a 32 byte block of memory starting from hexadecimal address 4000:

P 4000 401F (Print 32 bytes of memory starting from 4000 hex.)

Memory Edit

A memory edit facility can be used to change the value of a byte or bytes stored in read/write memory. Values are invariably specified in hexadecimal. A typical sequence of monitor commands to place a byte of 2A hex. into memory location 8000 hex. would take the form:
M 8000 4E – 2A (Modify the byte present at 8000 hex.)

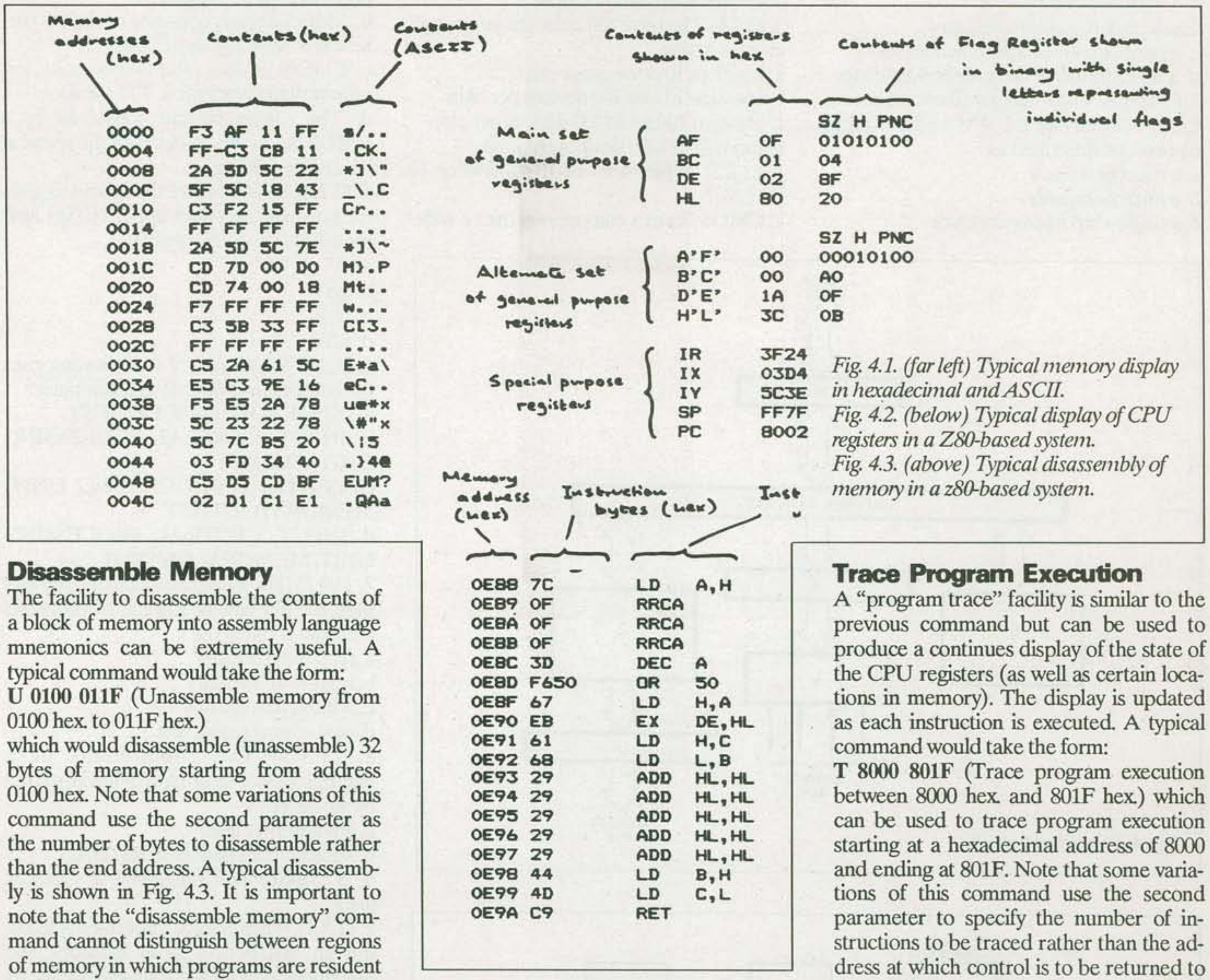
The system responds by displaying the hex. value (4E) of the byte currently present at address 8000H and the user then supplies the new value to be stored at the address. Note that there are many other variations of this command including those which preset (fill) a block of memory with a given byte value. A typical command to fill a 1K block of memory from 4000 hex. to 43FF hex. with a byte of 0F hex. would take the form:
F 4000 43FF 0F (Fill 1024 byte of memory starting from 4000 hex. with a byte value of 0F hex.)

CPU Register Display

The register display facility allows users to examine the contents of the CPU registers. No parameter follows the command which is typically entered using the single letter R. A typical register display for a Z80 CPU is shown in Fig. 4.2.

CPU Register Modify

The register modify command allows users to load the CPU registers with values prior to testing machine code routines. The command usually needs to be followed by a mnemonic for the register concerned, a typical example being:
S BC 00FF (Set the BC register pair to 00FF hex.)
which loads the BC register pair with the hexadecimal word. (Note that 00 appears in register B while FF hex. appears in register C). Again, there are many variations of this command.


Memory addresses (hex)
Contents (hex)
Contents (ASCII)
Contents of registers shown in hex
Contents of Flag Register shown in binary with single letters representing individual flags
Main set of general purpose registers
Alternate set of general purpose registers
Special purpose registers
Memory address (hex)
Instruction bytes (hex)
Inst

```
0000   F3 AF 11 FF   s/..
0004   FF-C3 CB 11   .CK.
0008   2A 5D 5C 22   *]\"
000C   5F 5C 18 43   _\.C
0010   C3 F2 15 FF   Cr..
0014   FF FF FF FF   ....
0018   2A 5D 5C 7E   *]\~
001C   CD 7D 00 D0   M}.P
0020   CD 74 00 18   Mt..
0024   F7 FF FF FF   w...
0028   C3 5B 33 FF   C[3.
002C   FF FF FF FF   ....
0030   C5 2A 61 5C   E*a\
0034   E5 C3 9E 16   eC..
0038   F5 E5 2A 78   ue*x
003C   5C 23 22 78   \#"x
0040   5C 7C B5 20   \|5
0044   03 FD 34 40   .}4@
0048   C5 D5 CD BF   EUM?
004C   02 D1 C1 E1   .QAa
```

```
                 SZ H PNC
AF     0F        01010100
BC     01        04
DE     02        BF
HL     80        20

                 SZ H PNC
A'F'   00        00010100
B'C'   00        A0
D'E'   1A        0F
H'L'   3C        0B

IR     3F24
IX     03D4
IY     5C3E
SP     FF7F
PC     8002
```

```
0E88 7C          LD    A,H
0E89 0F          RRCA
0E8A 0F          RRCA
0E8B 0F          RRCA
0E8C 3D          DEC   A
0E8D F650        OR    50
0E8F 67          LD    H,A
0E90 EB          EX    DE,HL
0E91 61          LD    H,C
0E92 68          LD    L,B
0E93 29          ADD   HL,HL
0E94 29          ADD   HL,HL
0E95 29          ADD   HL,HL
0E96 29          ADD   HL,HL
0E97 29          ADD   HL,HL
0E98 44          LD    B,H
0E99 4D          LD    C,L
0E9A C9          RET
```

Fig. 4.1. (far left) Typical memory display in hexadecimal and ASCII.
Fig. 4.2. (below) Typical display of CPU registers in a Z80-based system.
Fig. 4.3. (above) Typical disassembly of memory in a z80-based system.

Disassemble Memory

The facility to disassemble the contents of a block of memory into assembly language mnemonics can be extremely useful. A typical command would take the form:

U 0100 011F (Unassemble memory from 0100 hex. to 011F hex.)

which would disassemble (unassemble) 32 bytes of memory starting from address 0100 hex. Note that some variations of this command use the second parameter as the number of bytes to disassemble rather than the end address. A typical disassembly is shown in Fig. 4.3. It is important to note that the "disassemble memory" command cannot distinguish between regions of memory in which programs are resident and those which contain text or data. The result of using the command will be meaningless in the latter case.

Insert Breakpoints

Breakpoints are codes (usually a single byte) which are inserted into programs during testing or debugging. When a breakpoint is subsequently encountered, execution of the program is suspended and control is returned to the monitor program. This facility allows the user to examine the state of the microprocessor's registers and any relevant memory locations when a certain point is reached.

A typical command to insert a breakpoint takes the form:

B 801F (insert a Breakpoint at address 801F hex.)

which inserts a breakpoint at address 801F hex. Most monitors allow users to insert a number of breakpoints thus allowing for programs which use conditional branching.

Execute a Program

This command should be self-explanatory. The command usually expects a single parameter which is the hexadecimal address from which execution is to commence. Some monitors (including DEBUG) allow further parameters which constitute addresses at which breakpoints are to be placed. A typical command would take the form:

G 8000 (Go from address 8000 hex.)

This command sets the Instruction Pointer (or Program Counter) to 8000 hex. and commences program execution from that address. Note that this command is a little dangerous as, unless breakpoints have been appropriately placed, control may not be returned to the monitor program and the user may thus effectively lose control of the system.

Trace Program Execution

A "program trace" facility is similar to the previous command but can be used to produce a continues display of the state of the CPU registers (as well as certain locations in memory). The display is updated as each instruction is executed. A typical command would take the form:

T 8000 801F (Trace program execution between 8000 hex. and 801F hex.) which can be used to trace program execution starting at a hexadecimal address of 8000 and ending at 801F. Note that some variations of this command use the second parameter to specify the number of instructions to be traced rather than the address at which control is to be returned to the monitor program. The trace facility is sometimes known as a "single-step" facility.

First Written Assessment

By now, readers should also be adequately prepared for the first written assessment. This is a straightforward multiple choice test containing 22 questions, each of which is provided with four answers. Candidates should choose the answer which they think is correct and mark the Answer Sheet accordingly. Note that only ONE answer is correct in each case.

This test contains 22 multiple-choice questions. In order to pass you must answer a minimum of 16 of them correctly. The first number for each question is the test question number while the number in bracket relates to the module reference.

1. (1.1.1) A read-only memory (ROM) is best defined as

a. a volatile memory
b a form of read/write memory
c. a permanently programmed memory
d. a memory which can only be written to.

2. (1.1.2) A VLSI integrated circuit having CPU, internal clock, RAM and I/O ports can best be described as
a. a microprocessor
b. a microcomputer
c. a single-chip microprocessor
d. a single-chip microcomputer.

3. (1.2.1) The term SSI refers to integrated circuits having
a. less than 10 devices per chip
b. between 10 and 100 devices per chip
c. between 100 and 1000 devices per chip
d. more than 1000 devices per chip.

4. (1.2.2) Which one of the following is FALSE?
a. CMOS devices can operate over a wide range of supply voltages.
b. CMOS devices consume negligible current in a "standby" condition.
c. CMOS devices offer better noise immunity than comparable TTL devices.
d. The supply current consumed by a CMOS device decreases with the speed at which it is switching.

5. (1.2.3) Which one of the following gives the normally accepted upper voltage limit for a TTL low state (logic 0)?
a. 0V
b 0.8V
c. 2.0V
d. 5.0V

6. (1.3.1) Which one of the following gives the constituent elements of a computer?
a. INPUT, MEMORY, OUTPUT
b. INPUT, CENTRAL PROCESSING UNIT, OUTPUT
c. CENTRAL PROCESSING UNIT, MEMORY, OUTPUT
d. INPUT, CENTRAL PROCESSING UNIT, MEMORY, OUTPUT

7. (1.3.2) The computer element responsible for permanent storage of programs and data is called the
a. read only memory
b. read/write memory
c. arithmetic logic unit
d. central processing unit.

8. (1.3.3) The components of a microprocessor system are linked together by means of
a. input/output lines
b. address, data path (e.g. RS-232C)
d. a control bus (including a CLOCK line).

9. (1.3.4) When a microprocessor is fetching an instruction from memory, the operation code for the instruction appears on the
a. address bus
b. control bus
c. data bus
d. I/O bus

10. (2.1.1) the block marked 'X' in Fig. 4.4 represents the
a. data bus buffer Continued on page 48

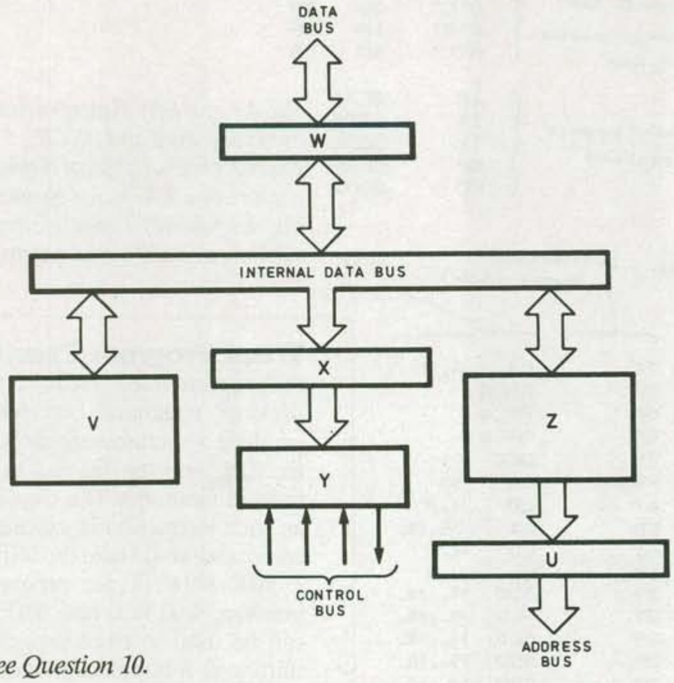


Fig. 4.4. See Question 10.

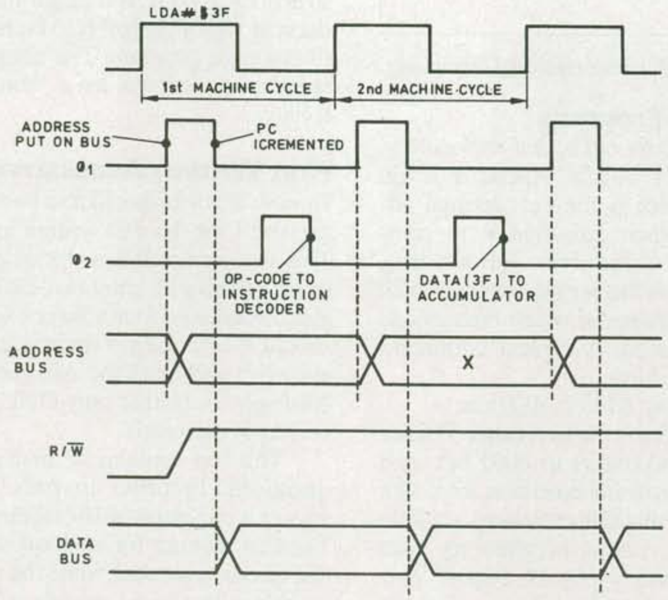


Fig. 4.5. See Question 21.

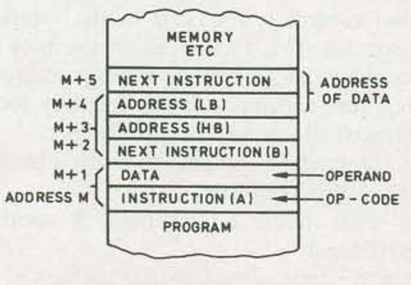


Fig. 4.6. See Question 22.

Microprocessors, Part 4 *Continued from page 12*

b. instruction decoder
c. instruction register
d. CPU registers (including Instruction Pointer).

11. (2.1.2) The instruction Register provides a means of
a. locating the next instruction in memory
b. locating the start address of a program in memory
c. storing an instruction while it is being decoded
d. storing the result produced when an instruction is executed.

12. (2.1.3) Which one of the following statement is TRUE?
a. Both RAM and ROM devices are connected to the address bus
b. neither RAM nor ROM devices are connected to the address bus
c. RAM devices are connected to the address bus while ROM devices are not connected to the address bus
d. ROM devices are connected to the address bus while RAM devices are not connected to the address bus.

13. (2.1.4) The clock in a microprocessor system provides
a. a common time reference
b. a control signal for read/write memory
c. a signal used within the address decoder
d. a means of determining the execution time of a program.

14. (2.1.5) The clock input to a microprocessor comprises
a. a sine wave signal of typically 1kHz to 4kHz
b. a square wave signal of typically 1kHz to 4kHz
c. a sine wave signal of typically 1MHz to 4MHz
d. a square wave signal of typically 1MHz to 4MHz.

15. (2.2.1) Which one of the following is an essential part of any microprocessor instruction?
a. Label
b. Address
c. Operand
d. Operation code.

16. (2.2.2) The first byte of a three-byte instruction comprises
a. a label
b. an address
c. an operand
d. an operation code.

17. (2.2.3) Which one of the following does NOT represent a data transfer instruction?
a. JMP
b. LDA
c. MOV
d. STA.

18. (2.2.4) In which one of the following addressing modes are there no data or address bytes present within an instruction?
a. Absolute
b. Extended
c. Implied
d. Immediate

19.(2.3.1) Which one of the following gives the correct sequence of events within the fetch-execute cycle of an instruction which loads the accumulator with an immediate data byte?
A = Decode the Operation Code
B = Increment the Instruction Pointer
C = Fetch Operand
D = Latch Operand into Accumulator
E = Fetch Operation Code
a. A E B C D
b. A B E D C
c. E B A C D
d. E C B A D

20. (2.3.2) Which one of the following gives the action which takes place in the instruction Pointer during the fetch-execute cycle?
a. Set to zero and then incremented as each instruction byte is fetched hence counting the total bytes in a program.
b. Set to the first address of the program and maintained constant so that the program can be restarted when a system RESET occurs.
c. Incremented during the fetch-execute cycle so that it eventually holds the address of the next instruction to be fetched.
d. Contains first the operation code and then each operand byte in strict sequence this allowing the instruction to be decoded.

21. (2.3.3.) The feature marked X in Fig. 4.5 represents
a. the address of the data byte
b. the data byte present on the bus
c. the address of the operation code
d. the operation code present on the bus

22. (2.3.4.) Fig. 4.6 shows two instructions (A and B) located within the memory of a microprocessor system. During the FIRST fetch cycle
a. the address M+1 is loaded with the operand
b. the address M is loaded with the operation code
c. the Instruction Pointer is loaded with the operand
d. the Instruction Pointer is loaded with the operation code.

Test Answers

1. d 2. d 3. a 4. d 5. b. 6. d 7. d 8. b 9. c 10. c 11. c 12. a 13. a 14. d 15. d 16. d 17. a 18. c 19. c 20. c 21. a 22. d ■